ENHANCING PRIVACY AND SECURITY IN EMBEDDED SYSTEMS THROUGH EDGE VPN



ARSHAD ULLAH
09-242242-002

BAHRIA UNIVERSITY ISLAMABAD

ENHANCING PRIVACY AND SECURITY IN EMBEDDED SYSTEMS THROUGH EDGE VPN



ARSHAD ULLAH
09-242242-002

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Master of Science in Computer Engineering

Department of Computer Engineering

BAHRIA UNIVERSITY ISLAMABAD

AUGUST 2026

© Arshad Ullah, 2026

MS-13
Thesis Completion Certificate

Student Name: **Arshad Ullah** Registration Number: **97826**
Program of Study: **Master of Science in Computer Engineering**
Thesis Title: **Enhancing Privacy and Security in Embedded Systems through Edge VPN**

It is to certify that the above student's thesis has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at **10%** that is within the permissible set by the HEC. for MS/MPhil/PhD.

I have also found the thesis in a format recognized by the BU for MS/MPhil/PhD thesis.

Principle Supervisor's Signature:_____
Principle Supervisor's Name: Dr. Syed Khawar Hussain Shah
August 20, 2026

MS-14A
Author's Declaration

I, **Arshad Ullah** hereby state that my MS thesis titled **"Enhancing Privacy and Security in Embedded Systems through Edge VPN"** is my own work and has not been submitted previously by me for taking any degree from **"Bahria University H-11, Islamabad"** or anywhere else in the country / world.

At any time if my statement is found to be incorrect even after my Graduate the university has the right to withdraw cancel my MS degree.

Arshad Ullah
09-242242-002
August 20, 2026

MS-14B
Plagiarism Undertaking

I, **Arshad Ullah** solemnly declare that research work presented in the thesis titled

Enhancing Privacy and Security in Embedded Systems through Edge VPN

is solely my research work with no significant contribution from any other person. Small contribution / help whenever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of Bahria University and the Higher Education Commission of Pakistan towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarised and any material used is properly referred / cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the university reserves the right to withdraw / revoke my MS degree and HEC and the university has the right to publish my name on HEC / University Website on which name of students who submitted plagiarised thesis are placed.

Arshad Ullah
09-242242-002
August 20, 2026

DEDICATION

This thesis is dedicated to my beloved family for their continuous support and encouragement during my academic career, to my friends and guides for their guidance and cooperation as I fulfilled my duties in my career and in academics, and to all working students who struggle to maintain balance between their careers and studies.

ACKNOWLEDGMENTS

I thank Almighty Allah for all His blessings and providing me strength and potential to complete my thesis.

I would like to pay gratitude to my supervisor Dr. Syed Khawar Hussain Shah for his continuous guidance, dedicated involvement and support throughout completing my thesis.

My sincere appreciation to all the Professors for their immense knowledge, without their teachings it wasn't possible to accomplish my work. My university fellows who always helped me whenever needed. I would like to thank my family for always supporting me.

ABSTRACT

The proliferation of Internet of Things (IoT) and the success of rich cloud services have pushed the horizon of a new computing paradigm, edge computing, which calls for processing the data at the edge of the network. Edge computing has the potential to address the concerns of response time requirement, battery life constraint, bandwidth cost saving, as well as data safety and privacy. In this paper, we introduce the definition of edge computing, followed by several case studies, ranging from cloud offloading to smart home and city, as well as collaborative edge to materialize the concept of edge computing. Finally, we present several challenges and opportunities in the field of edge computing and hope this paper will gain attention from the community and inspire more research in this direction. Specifically, the dataset has been generated using a purpose-built IoT/IIoT testbed with a large representative set of devices, sensors, protocols and cloud/edge configurations. The IoT data are generated from various IoT devices (more than 10 types) such as Low-cost digital sensors for sensing temperature and humidity, Ultrasonic sensor, Water level detection sensor, pH Sensor Meter, Soil Moisture sensor, Heart Rate Sensor, Flame Sensor, etc.). Furthermore, we identify and analyze fourteen attacks related to IoT and IIoT connectivity protocols, which are categorized into five threats, including, DoS/DDoS attacks, Information gathering, Man in the middle attacks, Injection attacks, and Malware attacks. In particular, we address the design of switches and control platforms with a focus on aspects such as resiliency, scalability, performance, security, and dependability Vas well as new opportunities for carrier transport networks and cloud providers. Last but not least, we analyze the position of SDN as a key enabler of a software-defined environment

TABLE OF CONTENTS

DECLARATION ii
DEDICATION iv
ACKNOWLEDGMENTS v
ABSTRACT vi
TABLE OF CONTENTS vii
LIST OF TABLES x
LIST OF FIGURES xi
LIST OF ABBREVIATIONS xiii
1 INTRODUCTION 1
1.1 Introduction 1
1.2 Motivation and Problem Description 2
1.3 Research Objectives 3
1.4 Research Contribution 3
1.5 Thesis Organization 5
2 LITERATURE REVIEW 6
2.1 Introduction 6
2.2 VPN Gateways for IoT and Embedded Devices 6
2.3 Combining Multiple Internet Links (VPN Bonding) 8
2.4 Routers, Firewalls, and Network Segmentation 8
2.5 Common Themes Across the Literature 9
2.6 What Literature Still Leaves Out 9
2.7 Comparison Table of Reviewed Papers 9
2.8 Conclusion 11
3 METHODOLOGY 12
3.1 System Architecture 12

3.2 VPN Gateway Configuration Procedure 14
3.2.1 Phase 1 — Establishing the Encrypted Tunnel 15
3.2.2 Phase 2 — Sharing the Tunnel with the LAN (Internet Connection Sharing) 15
3.3 Firewall Configuration Procedure (Huawei USG6370) 16
3.3.1 Accessing the Management Interface 16
3.3.2 Initial Dashboard Verification 16
3.3.3 Interface and Zone Configuration 16
3.4 Testing Procedure: Before and After VPN Deployment 17
3.4.1 Baseline Test (Before VPN Deployment) 17
3.5 Implementation Environment 17
3.6 Data Collection 18
3.7 Testing Sequence Summary 18
3.8 Ethical Considerations 20
3.9 Active Directory Deployment Procedure 20
3.10 SIEM (Splunk) Deployment Procedure 20
4 EXPERIMENTS AND RESULTS 22
4.1 Introduction 22
4.2 VPN Gateway Configuration Results 23
4.2.1 VPN Tunnel Establishment 23
4.2.2 Internet Connection Sharing Configuration 24
4.3 Firewall Access and Dashboard Verification 25
4.3.1 Firewall Login Interface 25
4.3.2 Dashboard and Device Health Verification 26
4.4 Security Policy Configuration Results 27
4.5 Threat and Attack Monitoring Results 28
4.6 Traffic and Destination Address Analysis 29
4.7 Network Interface and Zone Verification 30
4.8 URL Category and Website Access Monitoring 31

4.9 Blacklist and IP-Based Access Control 32
4.10 File Type and File Blocking Analysis 34
4.11 Active Directory Implementation and Organizational Structure 35
4.11.1 Active Directory Users and Computers Console 36
4.11.2 Right Pane Analysis: Built-in Security Groups in the Users Container 36
4.11.3 Threat Model: Active Directory Attack Paths and Mitigations 38
4.12 Integration with the Edge VPN Architecture 39
4.13 SIEM Solution Deployment (Splunk) 40
4.13.1 Splunk Installation and Access 41
4.13.2 Search Execution and Lookup-Based Dataset 41
4.13.3 Threat Detection Dataset and Sample Records 42
4.14 Before VPN Deployment Test 43
4.15 After VPN Deployment Test 43
4.16 Comparative Discussion: Before vs. After VPN Deployment 44
4.17 Discussion: Mapping Results to Research Questions 44
4.18 Limitations of the Experimental Results 45
4.19 Chapter Summary 46
5 CONCLUSION AND FUTURE WORK 48
5.1 Conclusion 48
5.2 Future Work 49
REFERENCES 51

LIST OF TABLES

2.1 Comparative Summary of Reviewed Literature 10

3.1 Implementation Environment 18

4.1 List of Figures Used in Chapter 4 22

4.2 Threat Type Occurrences 28

4.3 Top Destination Addresses by Traffic 30

4.4 URL Category Access Report 32

4.5 File Type Match Report 34

4.6 Recommended GPO Strategy for the theacme.io Domain 36

4.7 Built-in Security Groups in the Users Container with Security Significance 37

4.8 Active Directory Attack Threats and Mitigation Strategies 38

4.9 How Each Architecture Layer Protects the Active Directory Domain 40

4.10 Sample Threat Detection Records from security_example_data.csv 42

4.11 Network Attributes Before and After VPN Deployment 44

LIST OF FIGURES

3.1 Overall Research Methodology Flow 13

4.1 Windows Network Connections panel showing the “LAN”, “WAN”, and “ProtonVPN” (WireGuard Tunnel) adapters. 23

4.2 ProtonVPN adapter Properties, Sharing tab, showing Internet Connection Sharing enabled to the “LAN” adapter. 24

4.3 The login page for the web-based management for the Huawei USG6370. 25

4.4 Huawei USG6370 Dashboard with Health Overview, Device Information, Traffic History and Logs and Alarms. 26

4.5 All policies currently in the system and the number of times they have been hit, are displayed in Huawei USG6370 Security Policy List. 27

4.6 Threat Report, Threat Type view, showing Attack and Intrusion events over the preceding 24 hours. 28

4.7 Upstream traffic chart and destination-address ranking table, showing upstream, downstream, total traffic, and session counts. 29

4.8 Network to Interface list, showing all configured interfaces, their assigned zones, IP addresses, and VLAN/VXLAN settings. 30

4.9 URL Report, URL Category view, showing the number of times each category or user-defined site was accessed over 24 hours. 31

4.10 Security Protection to Blacklist page, showing the Blacklist function enabled and a list of manually added source/destination address entries. 32

4.11 Shows the type and volume of files that flow across the firewall over a 24 hour period via File Blocking Report and File Type view. 34

4.12 Active Directory Users and Computers (ADUC) console. 36

4.13 Splunk Enterprise 9.1.3 login page. 41

4.14 Splunk Search Job Inspector, execution cost breakdown for the lookup-based search (SID 1785172921.56). 41

4.15 Splunk dataset view of security_example_data.csv showing detected/mitigated threat records. 42

4.16 Public IP/geolocation lookup result captured before the VPN tunnel and ICS sharing were enabled. 43

4.17 Public IP/Geo location lookup result captured after the VPN tunnel and ICS sharing were fully configured and active. 43

LIST OF ABBREVIATIONS

AD	Active Directory
AD DS	Active Directory Domain Services
ADUC	Active Directory Users and Computers
AI	Artificial Intelligence
API	Application Programming Interface
BYOD	Bring Your Own Device
CS	Cyber Security
DC	Domain Controller
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DLL	Dynamic Link Library
DNS	Domain Name System
GPO	Group Policy Object
HTTPS	Hypertext Transfer Protocol Secure
ICS	Industrial Control Systems; Internet Connection Sharing
IDS	Intrusion Detection System
IoT	Internet of Things
IPS	Intrusion Prevention System
ISP	Internet Service Provider
KPI	Key Performance Indicator
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
ML	Machine Learning
NAS	Network Attached Storage
NIC	Network Interface Card
NTLM	NT LAN Manager

OU	Organizational Unit
PKI	Public Key Infrastructure
RDP	Remote Desktop Protocol
RODC	Read-Only Domain Controller
SDN	Software-Defined Networking
SIEM	Security Information and Event Management
SMB	Server Message Block
SOC	Security Operations Center
SOP	Standard Operating Procedure
SPN	Service Principal Name
SSH	Secure Shell
TLS	Transport Layer Security
VLAN	Virtual Local Area Network
VM	Virtual Machine
VPN	Virtual Private Network
VXLAN	Virtual Extensible LAN
WAN	Wide Area Network

CHAPTER 1

INTRODUCTION

1.1 Introduction

The Internet of Things (IoT) is a connected network of equipment that has the ability to communicate with each other and provide data to users via the Internet. The explosive growth of IoT in recent years is due in part to its broad applicability, scalability, and support for smart applications. The majority of IoT applications perform tasks in an automated fashion, with little or no interaction with humans.

Over the past decades, cloud computing [1, 2] has been largely developed and applied due to its high cost-efficiency and flexibility achieved through a combination of computing, storage, and network management functions that work in a centralized manner. Existing centralized cloud computing architecture is facing severe challenges owing to the fast development of the Internet of Things (IoT) and mobile Internet applications. For obtaining sophisticated applications, mobile devices connected to remote centralized cloud servers impose additional loads on both radio access networks (RANs) and backhaul networks that result in high latency [3]. Furthermore, with the progressing development of information technology, IoT is playing an important role in all aspects of current modern life [4]. With the advent of IoT, new challenges have arisen, such as stringent latency, capacity constraints, resource-constrained devices, uninterrupted services with intermittent connectivity, and enhanced security, which centralized cloud computing architecture cannot adequately satisfy this. IoT refers

to the modern interaction and communication network infrastructure of millions of physical objects (i.e., things) embedded with interconnected sensors/devices that can produce, collect, and exchange different data amongst themselves.

Therefore, more and more sensors and devices are being interconnected through IoT technology, and these sensors and devices will generate enormous amounts of data and require further processing and analytics, providing intelligence to both service providers and users. In conventional cloud computing, all data must be uploaded to centralized servers, and after computation, the results should be sent back to the sensors and devices. This process will require prohibitively high network bandwidth and the data transmission costs of bandwidth and resources.

1.2 Motivation and Problem Description

From a handful of well-documented failures involving embedded and IoT systems. Each one shows the same flaws from a different side: embedded devices are poorly Secure exactly where they meet the wider network.

- **Mirai Botnet (2016):** Attackers scanned for embedded devices that still used default or hard-coded credentials, took them over, and used them to launch some of the largest distributed denial-of-service (DDoS) attacks seen at the time. The devices gave their owners no way to rotate those credentials. The Mirai attacked demonstrated the consequences when lots of low-cost, poorly secured devices are left connected to the Internet.
- **Healthcare IoT breaches (2020-2023):** In multiple incidents, medical devices and hospital IoT equipment transmitted patient information without encryption or had outdated firmware, which enabled attackers to gain access to patient information. The common error was assuming

that the embedded medical device was trusted just because it was on the hospital network.

- **Industrial Control System (ICS) attacks:** Attackers have made it a habit of changing sensor data and actuator commands through unencrypted industrial protocols. An intrusion can become physical damage if it isn't encrypted or monitored at the operating-technology edge.

- All of these cases share a common theme: Cloud VPNs and standard perimeter firewalls never were designed to defend embedded nodes deep within a LAN. Most frequently, a cloud VPN will be used to secure a single client to a distant network.. It does little for the traffic running between a router, a firewall and the embedded devices behind them. A standard firewall can filter by port and address, but without a logging and correlation layer it cannot tell whether a strange event is a one-off or part of a wider intrusion.

1.3 Research Objectives

The objectives include the following:

- Identify practical techniques for protecting digital assets at the network edge that an organization without a dedicated security team could adopt.

1.4 Research Contribution

This thesis contributes a single Edge VPN-based security architecture for embedded systems, in which VPN tunneling, firewall enforcement and SIEM monitoring run as one coordinated edge deployment instead of separate tools that happen to share a network. The gap it addresses is that, as far as the reviewed literature shows, no existing study or implementation brings

Edge VPN, firewall and SIEM functions together into one scalable architecture designed for embedded systems that share a LAN with enterprise devices. The specific contributions are as follows:

1. Design and implementation of a dual-interface (dual-NIC) Edge VPN gateway positioned between the router and the firewall, encrypting all LAN traffic before it reaches the firewall.

2. A combined VPN and Firewall architecture that provides full packet tunnel monitoring, rather than treating encryption and filtering as separate, uncoordinated layers.

3. A local SIEM that collects and correlates logs from embedded devices, bringing monitoring that is normally reserved for large enterprise networks down to the edge.

4. A single encrypted tunnel, managed by the Edge VPN gateway, that puts privacy protection for all embedded devices under one point of control instead of relying on TLS/SSL implementations that differ from device to device.

5. Cryptographic protection applied at the edge before traffic reaches the cloud, whereas existing edge-computing security is mostly handled at the cloud endpoint.

6. Integration of SIEM, antivirus server and sandbox components with the Edge VPN gateway, giving end-to-end visibility across the embedded devices.

7. A practical, layered way of protecting digital assets at the edge, which organizations without a full security operations center can realistically adopt.

1.5 Thesis Organization

The thesis comprises five chapters. Chapter 2 includes the literature work and comparative analysis of scholarly work done by researchers in terms of Privacy and Security of Embedded systems through Edge VPN. Chapter 3 includes design and technical details of the proposed methodology, and Chapter 4 comprises the experimental setup, the detailed discussion of experiments, and their results. The last chapter, Chapter 5, concludes the research work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter reviews the research closest to the Edge VPN design proposed here. It covers the nine sources from the proposal and four newer studies from 2024 to 2026 that were added to keep the review current. The topics are VPN gateways for IoT, combining several internet links, routers and firewalls, security monitoring with SIEM, and small low-cost hardware such as the Raspberry Pi. A separate section then looks at the 2024–2026 papers on their own. The chapter closes with a comparison table of the reviewed papers and a short summary that ties the review back to the research gap stated in Chapter 1.

2.2 VPN Gateways for IoT and Embedded Devices

There are several studies that question how can the VPN technology that was developed for regular computer to computer connections be adapted to IoT and embedded devices. Abualigah et al. [1] proposed a hybrid VPN approach for IoT and compared with a basic VPN configuration, it was found that it was more effective against IoT threats. With this thesis, the point is that a VPN, built around IoT traffic, generally works better than a generic VPN.

Here, the choice of protocol is important. IPsec is secure and extensive, but consumes more processing power and increases the overhead for each

packet. WireGuard is lighter and faster, which is useful in small, low-power, embedded devices. This is the reason that WireGuard was selected for the gateway discussed in Chapter 3. Al-Khateeb [2] had a broader perspective, considering Bring Your Own Device (BYOD) environments, where personal devices are connected to a company network. The connection to this thesis is that embedded devices are often installed just like personal phones—fast and without much configuration or verification—and thus have vulnerabilities. Rahman et al. [3] paired VPN with Software-Defined Networking (SDN), so that a central controller manages the VPN connections automatically. That is flexibility, but comes with the requirement of an SDN controller as well as additional hardware and software that may not be easily justified in an environment as small as the 20-device testbed that has been used here. This is why this thesis is based on a simpler fixed dual-interface gateway. SDN stays on the list as an option for larger networks.

To enable two devices to agree on a secure channel without a central authority, Khan et al. [4] introduced a mathematical negotiation model, a Stackelberg game. One device suggests a security plan and the other responds in whatever manner is convenient, and they go back and forth until they're both onboard. This thesis is the opposite: a single VPN gateway which is trusted by everyone on the network. Having a central gateway makes it easier to manage and audit, however this means the whole network relies on that one gateway to be secure.

Canavese et al. [5] created a gateway dubbed the "IoT Proxy", which involves a VPN and machine learning intrusion protection device. Even recognize which device is truly attached to the network, without having to ask the IoT device to perform any additional computation. It's a good recent instance of VPN protection and automatic threat discovery in tandem at the edge, and a central tenet of this thesis: VPN and security monitoring will be most effective when synergized together, instead of being separate.

2.3 Combining Multiple Internet Links (VPN Bonding)

VPN bond is the act of establishing a safe tunnel using two or even more Web connections. These are not impossible as demonstrated by Zhang et al. [6] who have shown that it is possible to split data across multiple physical links, and that they can be reassembled at the VPN server, making the connection faster and more reliable, and if one link fails the tunnel will continue using the other links. This is crucial if a single connection cannot be relied on, such as in mobile, or remote locations. In this thesis the gateway is configured to have a single WAN connection which is primarily to simplify and reduce the costs of the first version. Later, the question of scaling up to additional devices and locations is raised (Research Question 2), in which bonding is still relevant.

2.4 Routers, Firewalls, and Network Segmentation

The WAN side of any network is the most vulnerable as it is directly connected to the internet [7] and so should be the place where strong security measures are implemented. Other researchers agree that the physical location where a network first interfaces with the outside world is typically where an attacker can more easily compromise the network [8]. The decision that was made in the proposal to deploy a hardware firewall behind the WAN facing interface rather than only the WAN facing software firewalls within each device is supported here.

The authors of Kumar et al. [8] also examined the management of VPNs in large-scale IoT networks and concluded that it is difficult to manage VPNs in the cloud scenario when there are hundreds or thousands of devices, and that it is better to move closer to the edge. That is directly related to Research Question 2 which is what happens to the design if it is scaled to 50 to 500 devices. In order to automatically select network routes based on existing traffic and potential hazards, Li and Wu [7] suggested the use of

AI. The concept is interesting, but is beyond the scope of the current design because it requires more computing than the small gateway here has.

2.5 Common Themes Across the Literature

Three themes are salient in these papers. Firstly, most studies consider the VPN, the firewall and monitoring separately and test them individually, and very few test them as a single system. Secondly, papers on IoT or embedded hardware [1, 9, 5] mostly address the issue of success or failure of a system and have little to say about long-term monitoring or management. Third, Monitoring and scaling papers typically consider large business networks with significant budgets [9, 8] and not small, low budget networks. This thesis is in the intersection of these three gaps.

2.6 What Literature Still Leaves Out

There are three gaps in literature. The first is scale: some of the studies were done on a simulated network or on an enterprise-size network, but not on a real, physically constructed small network. Combination is the second: in most studies, one layer (VPN, routing or detection) is improved, but it is not demonstrated how that improvement will play out with the other layers. The third is cost: Some solutions like the SDN controller in [3] and the AI-based routing in [7] require resources that smaller organizations may not possess.

2.7 Comparison Table of Reviewed Papers

Table 2.1 lists all eighteen reviewed papers side by side, with their main contribution and main limitation, using IEEE-style bracketed reference numbers.

Table 2.1: Comparative Summary of Reviewed Literature

Author(s)	Focus Area	Key Contribution	Limitation
Abualigah et al. [1]	Hybrid VPN for IoT	Mixed two VPN methods to better protect IoT devices.	Tested mostly on paper/protocol level, not on real hardware.
Al-Khateeb [2]	VPN for BYOD / industrial IoT	Looked at how VPNs can help with personal-device and industrial IoT risks.	More of a discussion paper; no working prototype.
Singh et al. [10]	VPN vs firewall comparison	Showed that VPN and firewall each protect against different problems.	Did not combine the two into one system.
Canavese et al. [5]	VPN + intrusion prevention gateway ("IoT Proxy")	Built a gateway that runs a VPN and a machine-learning-based intrusion prevention system together, and can even recognize which device is connecting.	No SIEM/log-correlation layer included; tested only in a lab.
Belarbi et al. [13]	Test dataset for IoT security research	Released a large, realistic dataset of normal and attack traffic from 78 emulated IoT devices, useful for testing security systems.	A dataset, not a security architecture by itself.
Gao et al. [11]	Lightweight intrusion detection at the edge	Built a small, fast model that can classify several attack types directly on edge hardware.	Focused only on detection, not on VPN or firewall integration.
Umar et al. [12]	Energy-efficient intrusion detection	Proposed a deep-learning detection model designed to use less power on edge devices.	Does not address VPN tunneling or firewall policy.
Hristov et al. [14]	Splunk SIEM for IoT	Splunk Enterprise for DDoS detection in IoT.	No Active Directory integration.
Nabil et al. [15]	SIEM analysis	Comprehensive SIEM features and best practices.	Survey paper only.
Azer et al. [16]	AD attack taxonomy	Classifies AD attacks by type and signature.	Detection only, no VPN architecture.
Kotlaba and Buchovecká [17]	Kerberoasting detection	ML detection of Kerberoasting.	Single attack type only.
NSA and CISA [18]	AD hardening	Government guidance on AD compromise detection.	Advisory, not an academic study.

2.8 Conclusion

The studies reviewed here show steady progress on each part of the proposed system: VPN gateways built for IoT, ways of combining internet links, stronger protection at the WAN entry point, SIEM-style monitoring for IoT networks, low-cost embedded hardware and, most recently, lightweight AI-based detection that runs on small devices. Even so, the research gap stated in Chapter 1 and the newer 2024–2026 papers point the same way: no published study combines VPN, firewall and SIEM in one edge-deployed system for a mix of embedded devices and ordinary office devices on the same network. That is the gap the Edge VPN architecture in this thesis is meant to fill.

CHAPTER 3

METHODOLOGY

This chapter describes how the Edge VPN-based security architecture introduced in Chapter 1 was built and tested. The overall method is the one set out in the proposal, a dual-interface VPN gateway placed between the internet and a managed firewall, but here the actual hardware, software and configuration steps are given. The sections follow the order in which the work was done. The VPN gateway was configured first, in two stages: establishing the encrypted tunnel, then sharing it with the LAN. The next was the firewall, which had its monitoring turned on. Finally, the whole configuration was validated by measuring the network behavior before and after the deployment of the VPN. This flow is illustrated in figure 3.1.

3.1 System Architecture

The architecture deployed basically adheres to the three tiered structure proposed in the project, which includes an internet facing VPN Gateway PC, a managed firewall and an internal LAN of client and embedded devices. The VPN Gateway PC is a Windows PC in the working configuration, which has two network adapters, as in the proposal.The adapter named “WAN” is used for connecting to the internet via the local ISP and the adapter named “LAN” is used to connect downstream to the trust-side interface of the Huawei firewall. This PC is running a ProtonVPN client, which will be using the new WireGuard protocol, and creating an outbound

encrypted tunnel to a ProtonVPN server in the Netherlands through the "WAN" adapter. This fulfills the proposal's requirement for a.

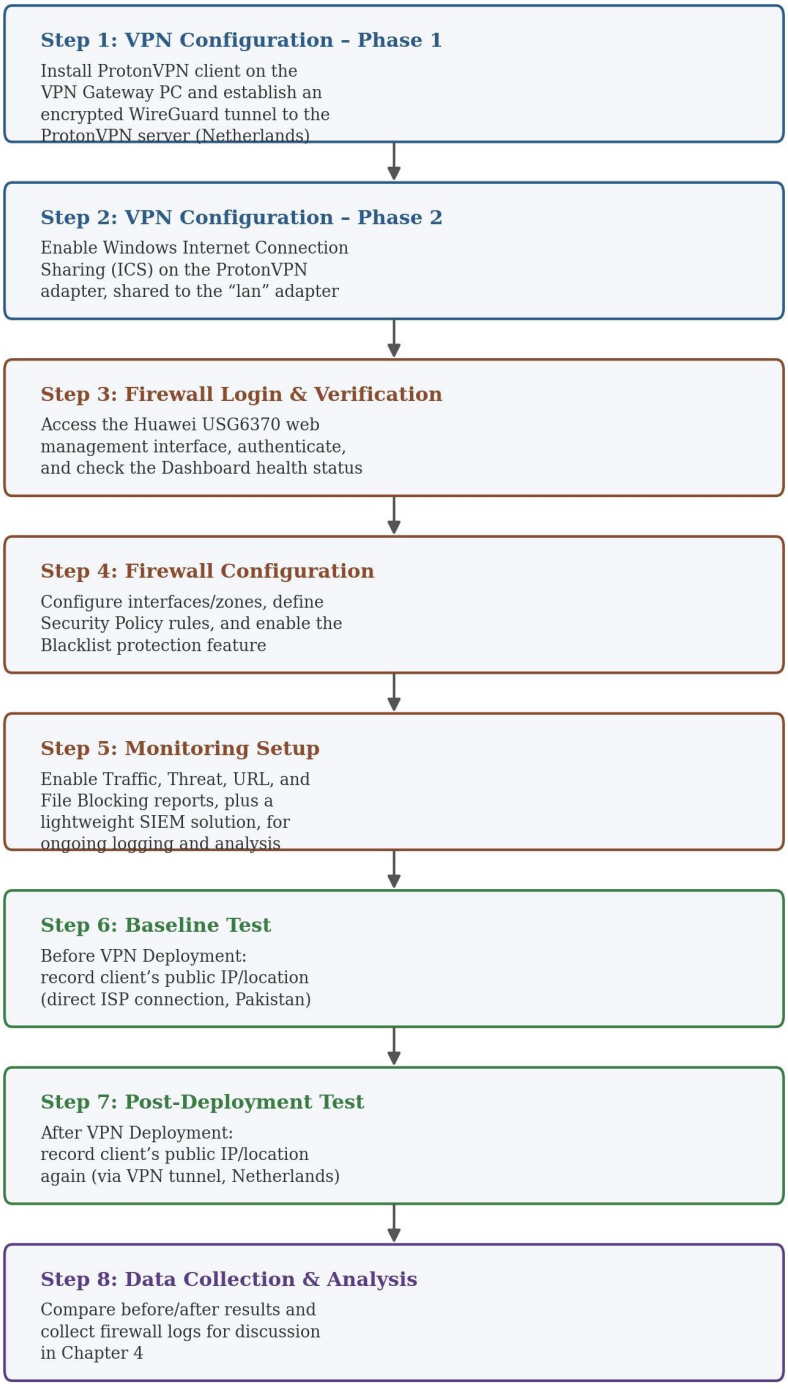


Figure 3.1: Overall Research Methodology Flow

requirement that the WAN NIC "connects to the cloud-based VPN server" and that the gateway "acts as the core encryption and decryption point for all data."

Rather than using iptables-based forwarding (the Linux-based alternative mentioned in the proposal), Windows Internet Connection Sharing (ICS) was used to share the ProtonVPN WireGuard tunnel with the "LAN" adapter. Once ICS is enabled, every device downstream of the "LAN" adapter—in this case, the Huawei firewall and everything behind it—has its traffic routed out through the encrypted VPN tunnel by default, without needing a VPN client installed on each individual device. This is consistent with the architectural goal, stated in Chapter 1, of protecting the whole LAN from a single, centrally managed trust point rather than securing each embedded device separately.

Downstream of the VPN Gateway PC sits the Huawei USG6370 firewall, matching the "Huawei USG-Series Managed Firewall" specified in the implementation environment of the proposal. Its management interface (GE0/0/0) is configured on the 192.168.10.1 network and placed in the trust zone; its internet-facing interface is configured in the untrust/public zone; and an internal VLAN (VLANif10, 10.10.0.5) segments the client and embedded-device network from the management network. This mirrors the proposal's description of the Huawei Managed Switch layer that provides "VLAN segmentation for embedded and user devices" and "isolated broadcast domains to prevent internal leakage", with segmentation implemented directly on the firewall's own interfaces rather than on a separate switch. Client workstations and embedded devices connect behind the firewall's trust zone, consistent with the original network topology diagram (Internet $\rightarrow$ VPN Gateway PC $\rightarrow$ Firewall $\rightarrow$ Switch $\rightarrow$ Embedded Systems/Clients) presented in the proposal.

3.2 VPN Gateway Configuration Procedure

The configuration of the VPN gateway was carried out in two distinct stages, here referred to as VPN Configuration Phase 1 and Phase 2,

which corresponded to the sequence in which the work was performed and documented.

3.2.1 Phase 1 — Establishing the Encrypted Tunnel

In the first stage, the ProtonVPN client was installed on the VPN Gateway PC and set to use the WireGuard protocol. Chapter 2 describes WireGuard as a lighter alternative to IPsec that suits gateways with limited processing headroom [5]. After the client connected, the Windows Network Connections panel displayed three active network adapters: "LAN", the internal network adapter listed as an "unidentified network" until the firewall was connected; "WAN", the ISP-facing adapter that shared with the other adapters; and "ProtonVPN", the WireGuard tunnel adapter itself. Seeing the ProtonVPN/WireGuard adapter active at this point confirmed that the encrypted tunnel was up before any LAN traffic was routed through it.

3.2.2 Phase 2 — Sharing the Tunnel with the LAN (Internet Connection Sharing)

In the second stage, Windows Internet Connection Sharing was enabled on the ProtonVPN adapter's properties, under the Sharing tab. Two options were enabled: "Allow other network users to connect through this computer's Internet connection," with the home networking connection set to "LAN," and "Allow other network users to control or disable the shared Internet connection." This step is what turns a single-machine VPN client into a shared gateway service for the whole LAN: once applied, any traffic arriving on the "LAN" adapter (in this case, from the Huawei firewall and everything behind it) is routed out through the encrypted ProtonVPN/WireGuard tunnel instead of directly through the "WAN" adapter.

This directly implements the proposal's Implementation Environment note that "the VPN gateway is configured to share the encrypted WAN tunnel with the LAN interface by sharing Internet Connection (ICS)."

3.3 Firewall Configuration Procedure (Huawei USG6370)

With the VPN gateway operational, the Huawei USG6370 firewall was configured and its monitoring features enabled. This was carried out in the following order.

3.3.1 Accessing the Management Interface

The firewall's web-based management interface was accessed over HTTPS, presenting the standard Huawei login page for the USG6370 model. An administrator account was used to authenticate. This front-end interface is the same one used for all subsequent configuration and monitoring steps described below.

3.3.2 Initial Dashboard Verification

The first thing to do after logging in was to see the health and baseline information of the device on the Dashboard, before making any change. It displayed the Health Overview (which is an overall score and does not contain any major or minor problems), Device Information (CPU, memory and disk usage, serial number, firmware version and running time) and Logs and Alarms (which is a list of system events like administrator logins and configuration changes). Starting from this check meant the firewall was in a healthy, known state when the VPN and policy work began.

3.3.3 Interface and Zone Configuration

The Network > Interface section was used to configure and verify the firewall's physical and virtual interfaces. Interfaces were assigned to security zones consistent with the trust/untrust model: GE0/0/0 (the management interface) and several LAN-facing interfaces were placed in the trust zone, while internet-facing interfaces such as GE1/0/6 were placed in the untrust zone with a public IP address. A VLAN interface (VLANif10) was configured in the trust zone to represent the internal client

and embedded-device segment. This interface configuration is what allows the firewall's security policy engine, described next, to distinguish between traffic originating from trusted internal devices and traffic arriving from the untrusted WAN side.

3.4 Testing Procedure: Before and After VPN Deployment

To check that the VPN gateway was doing what it was designed to do, masking and encrypting the LAN's outbound traffic, a simple before-and-after test was run from a client machine on the network, using a public IP-and-geolocation lookup tool.

3.4.1 Baseline Test (Before VPN Deployment)

Before the ProtonVPN tunnel and ICS sharing were enabled on the VPN Gateway PC, a client on the network was used to check its public-facing IPv4 address, IPv6 address, and country of origin. At this baseline stage, the client's traffic exited directly through the local ISP connection, and the lookup tool correctly reported a Pakistan-based IPv4 address with no VPN or proxy detected. This result was recorded as the "before deployment" baseline.

3.5 Implementation Environment

Table 3.1 summarizes the concrete hardware, software, and tools used in the implemented environment, updated from the proposal's original Implementation Environment table to reflect the specific products and configuration deployed.

3.6 Data Collection

Consistent with the three data sources identified in the proposal (network performance data, security log data, and operational data), data for this thesis was collected from the Huawei firewall's built-in Monitor and Report modules described in Section 3.3.6, plus the client-side IP/geolocation lookup results described in Section 3.4. Network performance and destination-traffic data were drawn from the Traffic Report and destination-address ranking views; security log data were drawn from the Threat Report, URL Report, File Blocking Report, and Blacklist views; and the before/after VPN deployment comparison served as a direct operational test of the gateway's core function. All of these results, captured as screenshots at the time of testing, are presented in Chapter 4.

Table 3.1: Implementation Environment

Component	**Specification / Role**
VPN Client Software	Proton desktop client, configured to use the WireGuard protocol
VPN Server	Proton exit server, Netherlands location
VPN Gateway PC	Windows-based dual-interface PC (Intel i5, 8GB RAM), with a "wan" adapter (ISP-facing) and a "lan" adapter (firewall-facing, TP-Link Gigabit PCI Express Adapter)
Tunnel-Sharing Method	Windows Internet Connection Sharing (ICS) on the Proton WireGuard adapter, shared to the "lan" adapter
Firewall	Huawei USG6370 next-generation firewall, managed through its web-based management interface
Firewall Interfaces	GE0/0/0 (management/trust, 192.168.10.1); GE1/0/6 (untrust/public, 50.50.0.15); Vlanif10 (trust, VLAN 10, 10.10.0.5) for the internal client segment
Embedded/Client Devices	Client workstations and embedded devices connected behind the firewall's trust zone
Monitoring Tools	Huawei firewall Monitor and Report modules (Traffic Report, Threat Report, URL Report, File Blocking Report); an online IP/geolocation lookup tool for VPN verification
Security Features Tested	Security Policy engine, Blacklist (manual and automatic), Attack Defense, application/URL filtering

3.7 Testing Sequence Summary

The following is a summary of the full sequence of events that were followed in implementing and testing; the order of events is as follows:

1. **VPN Configuration Phase 1 —** Install the ProtonVPN client on the VPN Gateway PC and connect it with WireGuard, then show that the tunnel is working in Windows Network connections panel.

2. **VPN Configuration Phase 2 —** Enable Internet Connection Sharing (ICS) on ProtonVPN adapter and share the Internet connection to the "LAN" adapter, which will serve as the Internet connection for the downstream devices.

3. **Firewall Login and Verification —** Open the Huawei USG6370 web management interface, log in and verify the health of the device on the Dashboard before changing it

4. **Firewall Configuration —** Create interfaces and zones, create security policy list (deny rule for torrenting, social networking and remote–access applications, and a default deny rule), and enable the Blacklist feature.

5. **Monitoring Setup —** Turn on the Traffic, Threat, URL and File Blocking reports to continuously record traffic and security events.

6. **Baseline Test (Before VPN Deployment) —** Note the Public IP Address and Country assigned to the client device when using an IP/geolocation lookup tool prior to enabling the VPN tunnel and the ICS sharing.

7. **Post-Deployment Test (After VPN Deployment)—** Do the lookup again after the tunnel and ICS sharing has been activated and compare the results with the baseline to ensure traffic is now passing through the tunnel. The results and screenshots in Chapter 4 are displayed in the same order: VPN Phase 1, VPN Phase 2, firewall logon and configuration and finally the before and after comparison.

3.8 Ethical Considerations

The research conducted in this proposal does not require personal or user data as it was done in the proposal. All collected data such as firewall traffic logs, threat and URL reports, file-type statistics, blacklist entries and IP/geolocation lookup results are network and system data and are only used for academic analysis. Identifying information such as IP addresses is only included in the text where necessary in Chapter 4 This is to demonstrate the functionality of the VPN gateway and no personally identifiable browsing content was collected or reported.

3.9 Active Directory Deployment Procedure

After installing the firewall and monitoring in Sections 3.3 and 3.4, a Windows Active Directory (AD) domain was installed behind the Edge VPN gateway and firewall to provide centralized identity and access management.. AD was chosen because it is still the dominant identity platform in Windows-based enterprise networks, and because centralized authentication is a prerequisite for enforcing the group-based access policies referred to in Chapter 1 [19, 16].

The deployment procedure followed the standard promotion sequence for a new AD forest and domain:

1. **Server Role Installation —** The Active Directory Domain Services (AD DS) role was installed on a Windows Server host connected to the trust-zone VLAN (VLANif10) described in Section 3.1, placing the domain controller behind the firewall's trust boundary rather than exposing it to the untrust zone.

3.10 SIEM (Splunk) Deployment Procedure

Section 3.3.6 noted that the firewall's own built-in Monitor and Report modules were used as the primary monitoring mechanism, in place of

the external Wazuh/Splunk SIEM originally listed as an implementation-environment option. This section describes the procedure subsequently followed to deploy that external SIEM layer using Splunk Enterprise, addressing that gap.

1. **Installation —** As discussed in Chapter 1, SIEM log collection is placed in the edge network, alongside the VPN gateway and firewall, instead of forwarding them to a third-party cloud SIEM, which made installing Splunk Enterprise 9.1.3 within the trust-zone network the logical choice.

2. **Authentication Setup —** An administrator account was configured for the Splunk instance, following the standard first-time sign-in procedure for a new Splunk Enterprise installation [14].

This SIEM layer is intended to eventually ingest the same Threat, Traffic Destination, URL, and File Blocking data already collected by the firewall in Sections 4.5–4.10, so that they can be correlated inside Splunk rather than viewed only through the firewall's own reporting console—directly addressing the SIEM-related limitation raised in Section 4.18 and the corresponding Future Work item in Chapter 5. The results of the validation deployment described above are presented in Section 4.13.

CHAPTER 4

EXPERIMENTS AND RESULTS

4.1 Introduction

This chapter presents the results of implementing and testing the Edge VPN-based security architecture described in Chapters 1 through 3. It follows the order of the implementation. First comes the VPN gateway configuration, with the encrypted tunnel and its sharing to the LAN. Then come the firewall login and initial checks, the firewall's security policy and monitoring configuration, and the reports the firewall produced over a 24-hour monitoring window. The last part is the before-and-after VPN deployment comparison, which tests the gateway's core security function directly. Each figure comes with a short discussion of what it shows, how it was obtained and what it says about the architecture.

Table 4.1: List of Figures Used in Chapter 4

Figure	Section	Summary
4.7	4.6	Traffic destination ranking by upstream, downstream, total traffic, and sessions.
4.8	4.7	Interface list showing trust and untrust zone assignments.
4.9	4.8	URL Report showing access to URL categories.
4.10	4.9	Blacklist configuration and manually blocked addresses.
4.11	4.10	File Blocking Report showing file types crossing the firewall.
4.12	4.11.1	Active Directory Users and Computers console for the theacme.io domain.
4.13	4.13.1	Splunk Enterprise login page.
4.14	4.13.2	Splunk Search Job Inspector execution-cost breakdown.
4.15	4.13.3	Splunk dataset view of detected/mitigated threat records.
4.16	4.14	Baseline public IP/location before VPN deployment.
4.17	4.15	Public IP/location after VPN deployment.

Note that the Threat, Traffic Destination, URL and File Blocking reports (Figures 4.6, 4.7, 4.9 and 4.11) all cover the same 24-hour window, from about 2026/06/23 12:20 to 2026/06/24 12:40. Reading them side by side, and not one at a time, is what reveals the pattern discussed in Section 4.19: a late-window peak in activity that shows up separately in the traffic volume, URL access and file-type charts.

4.2 VPN Gateway Configuration Results

This section presents the results of VPN Configuration Phases 1 and 2 described in Chapter 3, Section 3.2. These two figures together confirm that the encrypted WireGuard tunnel was successfully established and then correctly shared with the internal LAN, which is the foundation on which the rest of the architecture depends.

4.2.1 VPN Tunnel Establishment

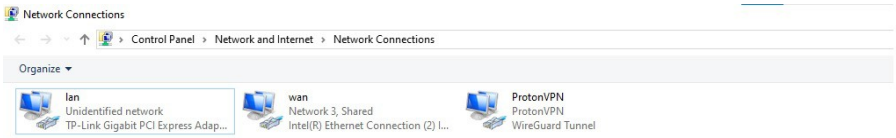


Figure 4.1: Windows Network Connections panel showing the “LAN”, “WAN”, and “ProtonVPN” (WireGuard Tunnel) adapters.

Figure 4.1 shows the Windows Network Connections panel on the VPN Gateway PC after the ProtonVPN client was configured to use the WireGuard protocol, as described in Section 3.2.1. Three adapters are visible: “LAN”, shown as an unidentified network at this stage because no downstream device had yet negotiated an IP configuration from it; “WAN”, labeled “Network 3, Shared”, which is the adapter connected to the ISP; and “ProtonVPN”, explicitly labeled “WireGuard Tunnel”, confirming that the VPN client is using the WireGuard protocol rather than OpenVPN or IPsec.

The presence of an active ProtonVPN/WireGuard adapter at this stage is the key evidence for this figure: it confirms that Phase 1 of the VPN configuration was completed successfully before any attempt was made to share the tunnel with the LAN in Phase 2. Without this adapter appearing and remaining connected, none of the subsequent steps, sharing the tunnel, routing firewall traffic through it, or observing a masked IP address in Section 4.15 would be possible.

4.2.2 Internet Connection Sharing Configuration

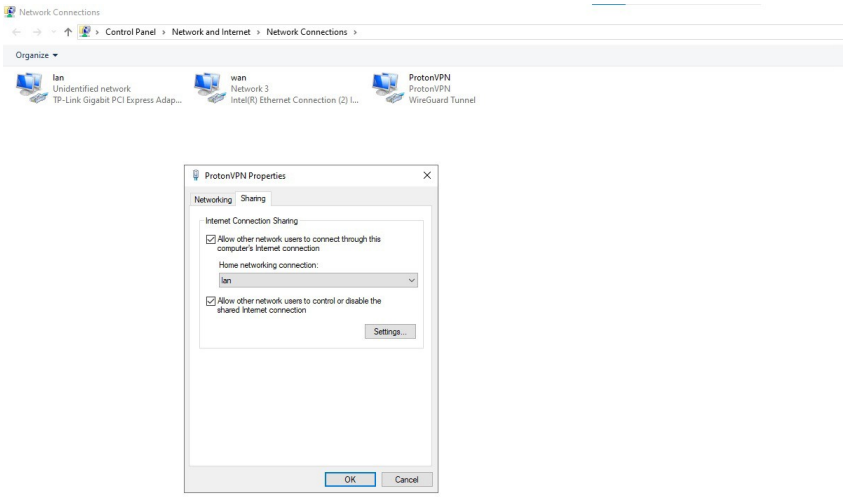


Figure 4.2: ProtonVPN adapter Properties, Sharing tab, showing Internet Connection Sharing enabled to the “LAN” adapter.

Figure 4.2 shows the Sharing tab of the ProtonVPN adapter’s Properties dialog, corresponding to VPN Configuration Phase 2 described in Section 3.2.2. Two settings are enabled: “Allow other network users to connect through this computer’s Internet connection”, with the Home networking connection explicitly set to “LAN”, and “Allow other network users to control or disable the shared Internet connection”.

This is the exact configuration step that turns a single-machine VPN client into a shared gateway service for the whole network. Once applied, Windows automatically begins routing all traffic arriving on the “LAN” adapter out through the ProtonVPN/WireGuard tunnel shown in Figure 4.1, instead of directly through the “WAN” adapter. the network from spoofing

attacks and regulate the entry of unauthorized traffic. The network from spoofing attacks and control of entry of unauthorized traffic is the one configuration change discussed in the proposed architecture (Chapter 1, Research Question 1). Installing a VPN client on each individual device is not necessary, throughout an entire downstream network from a central point

4.3 Firewall Access and Dashboard Verification

This section is about how to log in to and verify the Huawei USG6370 firewall, which are covered in Sections 3.3.1 and 3.3.2 of the methodology.

4.3.1 Firewall Login Interface

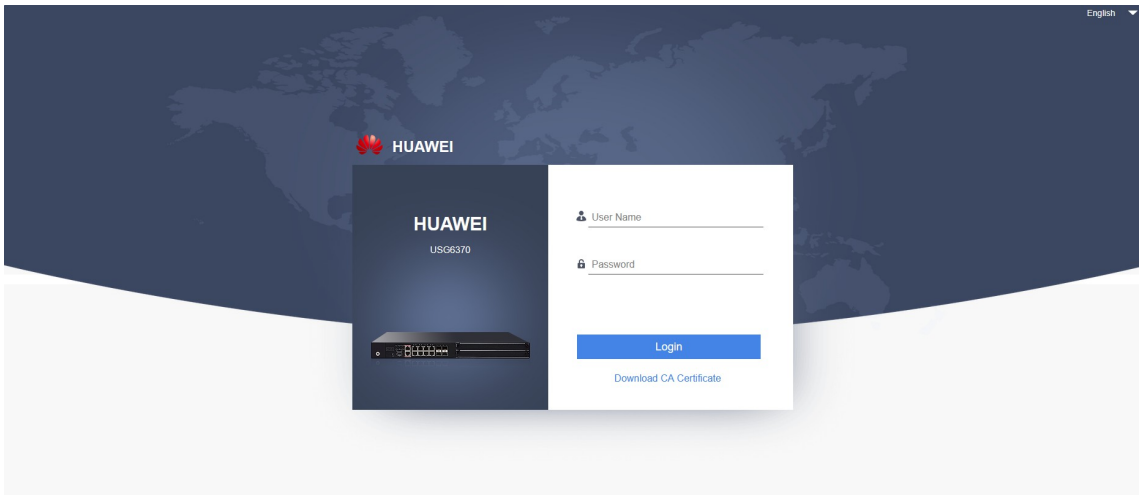


Figure 4.3: The login page for the web-based management for the Huawei USG6370.

The USG6370's web management interface login page is in Figure 4.3, over HTTPS. It specifies the exact model, provides a standard username/password type form and includes a downloadable device CA certificate to access the device via a secure browser. All other configuration and monitoring steps in this chapter begin here and all later figures were obtained after being logged on via this login page through admin account.

4.3.2 Dashboard and Device Health Verification

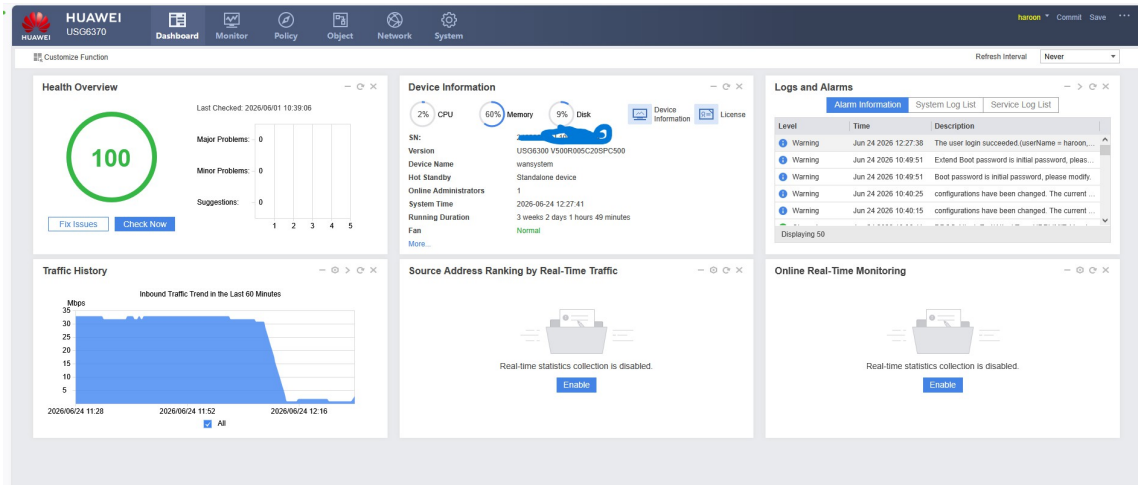


Figure 4.4: Huawei USG6370 Dashboard with Health Overview, Device Information, Traffic History and Logs and Alarms.

The first health check (described in Section 3.3.2) is shown in the firewall dashboard immediately after logging in, as in Figure 4.4. No problems were reported before any of the changes to the configuration were made, the Health Overview score was 100. The CPU utilization is 2%, Memory utilization is 60%, and Disk utilization is 9%. Also, it indicates the firmware version (USG6300 V500R005C20SPC500) and duration of three weeks, two days and ~1h49m, and it confirms the unit is running standalone, not as a hot-standby pair.

Inbound traffic is normal and peaks at around 30-35 Mbps and then makes a sudden drop on the Traffic History panel, 60 minutes of traffic history. The Logs and Alarms panel displays details of recent events in the system, including successful logon of an administrator, and a few configuration-change alerts that correspond to the configuration work done immediately following in Sections 4.4 onwards. It is common practice in network security devices to review the dashboard prior to performing any action, and Any results later in this chapter may be attributed to the configuration changes as a result of this study and not to any defect that was present.

4.4 Security Policy Configuration Results

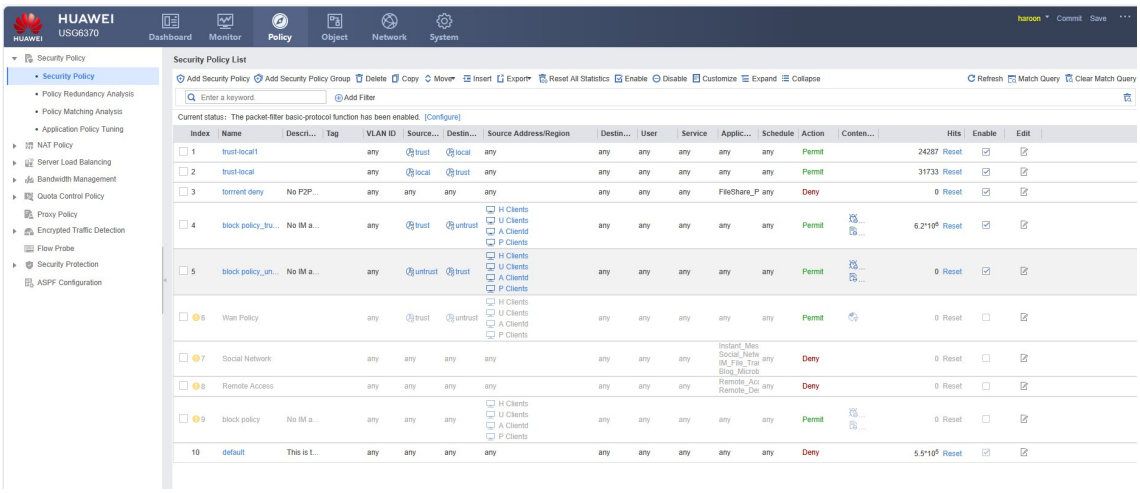


Figure 4.5: All policies currently in the system and the number of times they have been hit, are displayed in Huawei USG6370 Security Policy List.

The Security Policy List (Figure 4.5) is the output of the policy step in Section 3.3.4. It contains 10 policies that will be applied from top to bottom. The order is important since the firewall processes policies in the order they were supplied and applies the first one that matches. Policies 1 and 2 (“trust-local1” and “trust-local”) allow general traffic to flow between the trust and local zones, and at the time of capture they had hit 24,287 and 31,733 times, respectively, so that the general traffic was flowing as expected.

The 3 (“torrrent deny”) policy explicitly blocks traffic for peer-to-peer file-sharing. Policies 4 and 5 are general policies that apply to the groups of clients defined in the network (H Clients, U Clients, A Clientd, and P Clients) and policy 4 alone has had more than 6.2 million hits, indicating the amount of general trust-to-untrust traffic that flows through the gateway. Policy 6 (“Wan Policy”) permits general WAN-facing traffic but is shown as disabled (unchecked) at capture time. Policies 7 and 8 explicitly deny social-networking/instant-messaging applications and remote-access applications respectively, directly implementing the layered, application-aware filtering described in Chapter 1. Policy 9 is a broad permit rule for the same client groups, and Policy 10, named simply “default”, is the catch-all deny-all

rule with a description confirming that it is the default policy; it had already matched roughly 550,000 times at the point of capture, showing that a substantial volume of traffic not covered by any more specific rule correctly fell through to and was blocked by the final default-deny rule.

Taken together, this policy list demonstrates the layered security approach described in Chapter 1: broad internal trust is permitted, specific known-risky application categories (torrenting, social media, remote access) are explicitly denied, and anything left over is blocked by default rather than silently permitted, which is the safer of the two possible default postures for a security-focused deployment.

4.5 Threat and Attack Monitoring Results

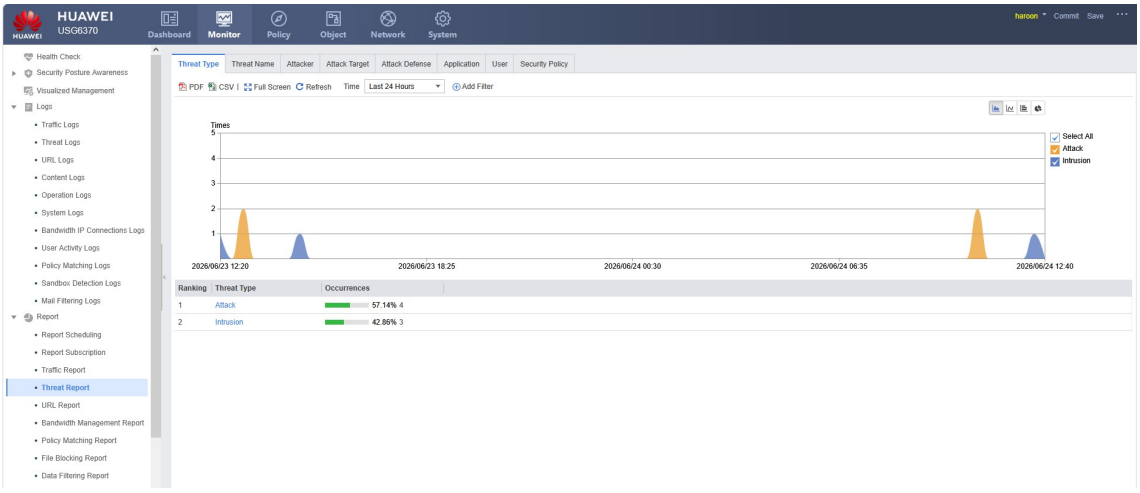


Figure 4.6: Threat Report, Threat Type view, showing Attack and Intrusion events over the preceding 24 hours.

Table 4.2: Threat Type Occurrences

Ranking	Threat Type	Occurrence Percentage	Occurrences
1	Attack	57.14%	4
2	Intrusion	42.86%	3

Figure 4.6 is the firewall's Threat Report, in the Threat Type view for the previous 24 hours, set up as described in Section 3.3.6. There are four peaks in the timeline, two of them of type Attack and the other two of type

Intrusion. They are distributed throughout the window, and not clustered on a single moment, indicating real and intermittent threat activity, not a one time event due to misconfiguration or false positive.

The summary table below the chart repeated in Table 4.2 indicates that there were 7 threat events detected, with 4 events being Attack events, and 3 events being Intrusion events, for a total of 57.14 percent (4 events) for Attack, and 42.86 percent (3 events) for Intrusion. Not many - as expected for a controlled academic testbed and not a live production network. Important is that both categories are displayed: the firewall's threat detection engine was performing an evaluation of the traffic crossing the firewall.

Edge VPN gateway, so as not to obscure the threats signatures from the firewall downstream. This is in line with the aim of having VPN protection and active threat detection at the same edge point (Canavese et al. [5]; Gao et al. [11]) rather than as back-to-back layers.

4.6 Traffic and Destination Address Analysis

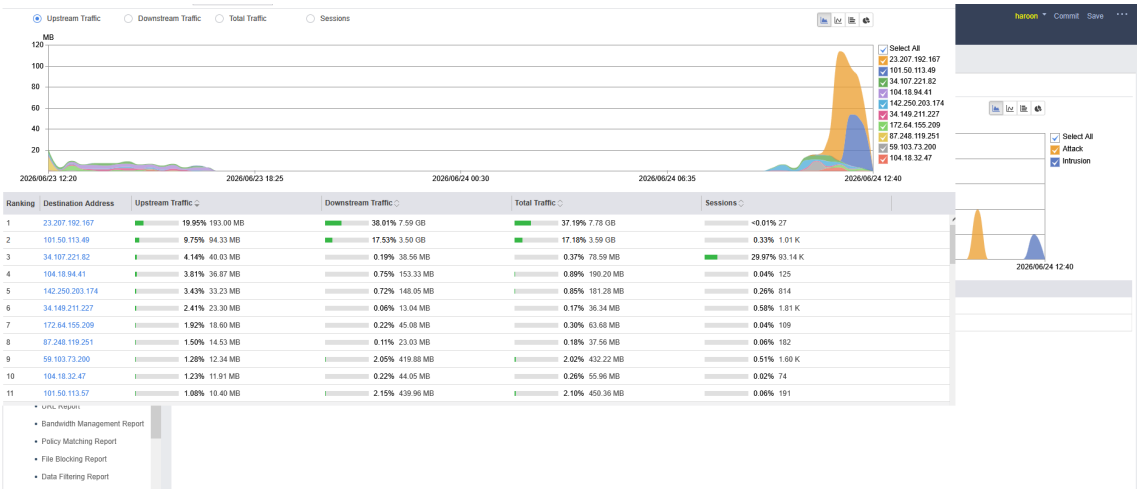


Figure 4.7: Upstream traffic chart and destination-address ranking table, showing upstream, downstream, total traffic, and session counts.

Figure 4.7 shows the destination-address ranking report, viewed by Upstream Traffic, over the same 24-hour window as Figure 4.6. The chart shows traffic climbing steadily through the monitoring period and peaking

sharply near the end of the window, reaching close to 120 MB at the highest point, before the ranking table below breaks that total traffic down by individual destination IP address.

Table 4.3: Top Destination Addresses by Traffic

Rank	Destination	Upstream	Downstream	Total	Sessions
1	23.207.192.167	193.00 MB (19.95%)	7.59 GB (38.01%)	7.78 GB (37.19%)	27
2	101.50.113.49	94.33 MB (9.75%)	3.50 GB (17.53%)	3.59 GB (17.18%)	1.01 K
3	34.107.221.82	40.03 MB (4.14%)	38.56 MB (0.19%)	78.59 MB (0.37%)	93.14 K
4	104.18.94.41	36.87 MB (3.81%)	153.33 MB (0.75%)	190.20 MB (0.89%)	125
5	142.250.203.174	33.23 MB (3.43%)	148.05 MB (0.72%)	181.28 MB (0.85%)	814
6–11	(6 further destinations)	2.41–1.08%	0.06–2.15%	0.17–2.10%	74–1.81 K

4.7 Network Interface and Zone Verification

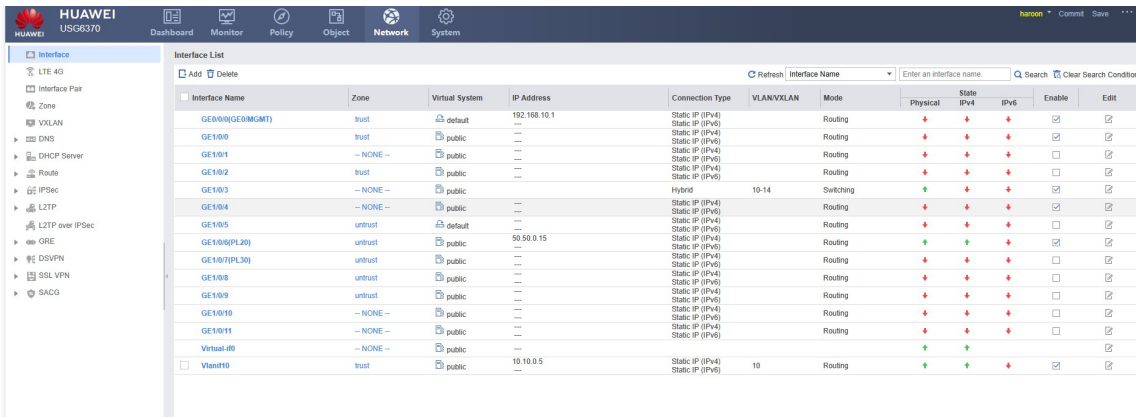


Figure 4.8: Network to Interface list, showing all configured interfaces, their assigned zones, IP addresses, and VLAN/VXLAN settings.

Figure 4.8 shows the full Interface List configured on the firewall, corresponding to the interface and zone configuration step described in Section 3.3.3. The management interface, GE0/0/0, is assigned to the trust zone with IP address 192.168.10.1. Several GE1/0/x interfaces are assigned to the trust zone (used for internal client connectivity), while others, including GE1/0/5 through GE1/0/11, are assigned to the untrust zone, consistent

with their role as WAN-facing or otherwise external-facing ports. GE1/0/6, labeled PL20, is configured with a public IP address of 50.50.0.15 and shows an active physical and IPv4 state (green up arrows), confirming it is the interface actually carrying live WAN traffic at the time of capture.

GE1/0/3 is configured in Hybrid mode with a VLAN range of 10–14 and a Switching connection type, rather than Routing, indicating it operates as a Layer 2 trunk carrying multiple VLANs. Vlanif10, the Layer 3 VLAN interface for VLAN 10, is assigned to the trust zone with IP address 10.10.0.5 and shows active physical, IPv4, and IPv6 state, confirming that the internal client segment (VLAN 10) referenced throughout Chapter 3 is correctly configured and operational.

4.8 URL Category and Website Access Monitoring

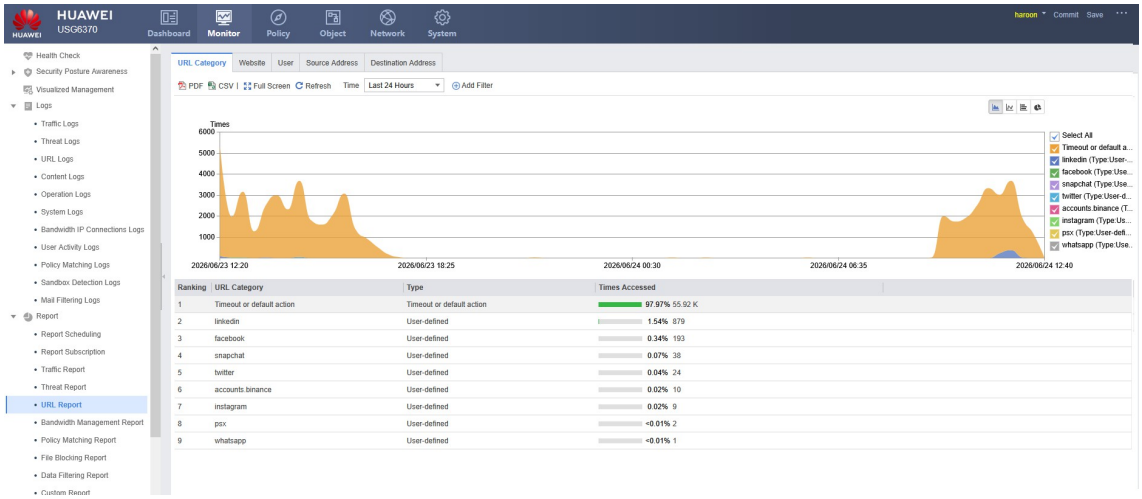


Figure 4.9: URL Report, URL Category view, showing the number of times each category or user-defined site was accessed over 24 hours.

Table 4.4: URL Category Access Report

Rank	URL Category	Type	Times Accessed
1	Timeout or default action	Timeout or default action	97.97% — 55.92 K
2	linkedin	User-defined	1.54% — 879
3	facebook	User-defined	0.34% — 193
4	snapchat	User-defined	0.07% — 38
5	twitter	User-defined	0.04% — 24
6	accounts.binance	User-defined	0.02% — 10
7	instagram	User-defined	0.02% — 9
8	psx	User-defined	<0.01% — 2
9	whatsapp	User-defined	<0.01% — 1

Figure 4.9 shows the URL Report, filtered to the URL Category view, over the same 24-hour monitoring window. The chart shows two prominent peaks, one around the middle of the window and a larger one toward the end, closely mirroring the shape of the traffic and file-type charts discussed elsewhere in this chapter, which suggests these peaks correspond to the same overall period of heavier network activity rather than being specific to URL filtering alone.

4.9 Blacklist and IP-Based Access Control

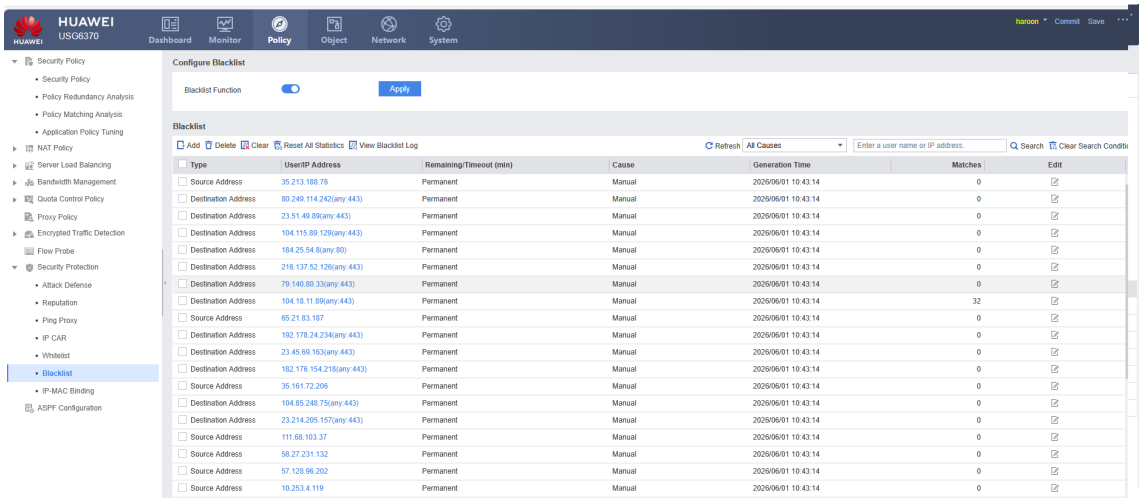


Figure 4.10: Security Protection to Blacklist page, showing the Blacklist function enabled and a list of manually added source/destination address entries.

Figure 4.10 is the firewall's Blacklist configuration page, the Security Protection step from Section 3.3.5. The Blacklist Function toggle is turned

on and the list below it contains 18 entries, of which 9 are of type Source Address and 9 are of type Destination Address. All entries are added in such a way that they are not automatically generated as temporary blocks, but rather added by hand with the Remaining/Timeout value of “Permanent” and the Cause of “Manual”. They all have the same generation time 2026/06/01 10:43:14 to allow to fill the blacklist for a single administrative session.

The vast majority have a Matches of 0: the listed address had not attempted to communicate through the firewall since it was blacklisted. The blacklist is only causing the live attack traffic to be sent to a blacklist and is not preventing the traffic from being received during the capture window for those addresses. There’s one thing that’s different about one entry. 32 matches were found for destination address 104.18.11.89 on port 443, indicating that traffic to 104.18.11.89 on port 443 occurred many times in the monitoring period, but was blocked each time. It’s solid evidence that the blacklist was doing more than just sitting configured – it was blocking real connections.

It is separate from the complete list of Security Policies in Figure 4.5, and performs significantly faster than that list, providing a second, lower-latency, address-based policy filtering layer to the zone-and application-based policy engine of the architecture.

4.10 File Type and File Blocking Analysis

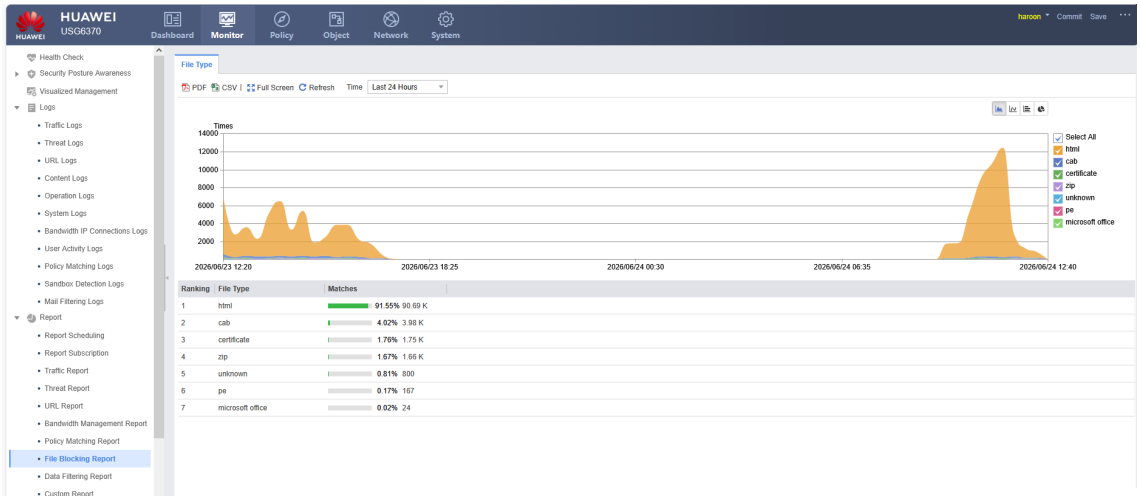


Figure 4.11: Shows the type and volume of files that flow across the firewall over a 24 hour period via File Blocking Report and File Type view.

Table 4.5: File Type Match Report

Rank	File Type	Percentage	Matches
1	html	91.55%	90.69 K
2	cab	4.02%	3.98 K
3	certificate	1.76%	1.75 K
4	zip	1.67%	1.66 K
5	unknown	0.81%	800
6	pe (executable)	0.17%	167
7	microsoft office	0.02%	24

The File Blocking Report listed in the File Type view is for the same 24-hour period as the other monitoring reports, and is shown in Figure 4.11. The peak in the late window, though, is again big, around 14,000 matches, with the same The shape as the URL and traffic reports. Every one of these reports seem to be tracking the same timeframe, and that's the case here too, as they all take a different perspective on traffic—such as URL category or file type. Table 4.5 lists the HTML files as the largest category at 91.55% (90,690 matches), as is normally the case given the fact that almost every web page loaded in normal web browsing is an HTML file. The other types are more important for security. Common containers for legitimate software and malware are cab files (4.02%) and zip archives

(1.67%). Certificates (1.76%) are TLS/HTTPS handshakes required to establish TLS/HTTPS connections. The firewall also caught and logged 167 portable-executable (“pe”) files directly, which means that it can flag on transfers of executables and not only on transfers of generic binary files. Another 800 files (0.81%) were recorded as “unknown” meaning that the firewall was unable to identify the type of file based on the signatures in its database. That is a small percentage but is noteworthy in any expansion or improvement of this work as a way to circumvent type-based filtering rules.

4.11 Active Directory Implementation and Organizational Structure

Active Directory Domain Services (AD DS) is Microsoft’s directory service. It stores information about network objects, meaning users, computers, groups and policies, and makes it available to administrators and users across an organization [23]. Deploying AD behind the VPN gateway and firewall turns the network from a set of independently managed endpoints into a centrally governed domain run by policy. This section describes the AD implementation carried out in this work in full. It starts with the console view in Figure 4.12, then moves through the domain structure, the security group design, the Group Policy implications, and the security and threat model that the directory layer brings to the architecture.

4.11.1 Active Directory Users and Computers Console

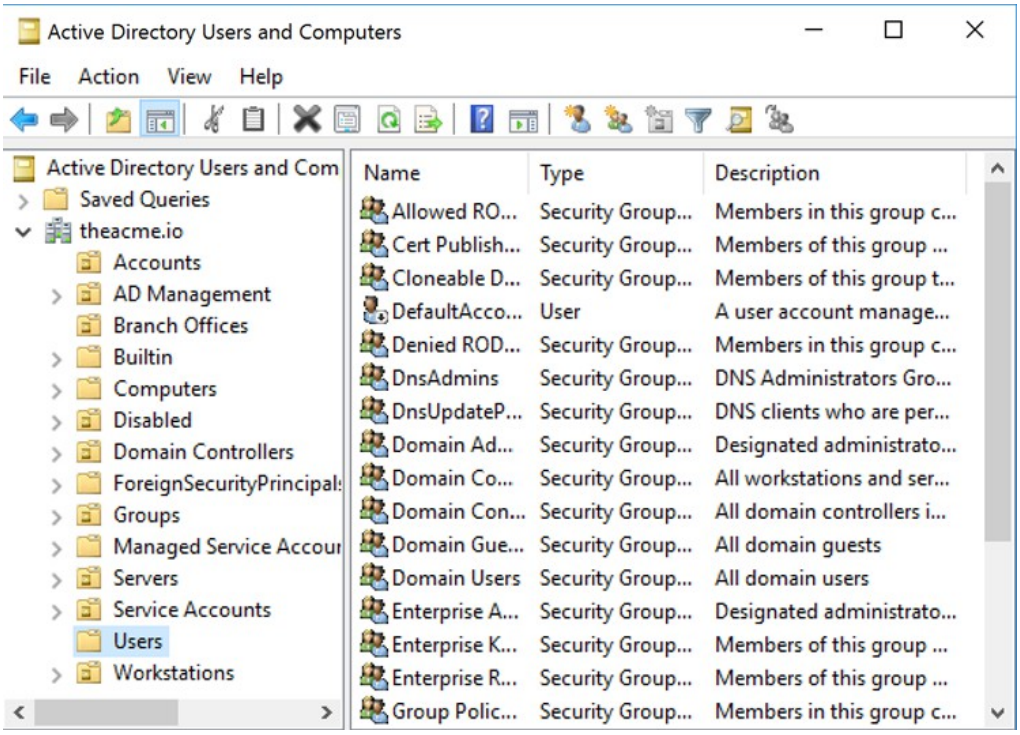


Figure 4.12: Active Directory Users and Computers (ADUC) console.

The OU structure shown in Figure 4.12 is specifically designed to support a tiered Group Policy strategy, as shown in Table 4.6.

Table 4.6: Recommended GPO Strategy for the theacme.io Domain [21, 24]

GPO Name	Linked OU	Enforced Settings
Default Domain Policy	Domain root	Password policy, account lockout, and Kerberos policy only.
Default DC Policy	Domain Controllers	Elevated audit and security settings for domain controllers.
Server Hardening GPO	Servers	Stricter auditing, disabling USB storage, and enforcing Network Level Authentication (NLA) for RDP.
Workstation Baseline GPO	Workstations	Screen lock, Windows Update configuration, and AppLocker rules.
Service Account GPO	Service Accounts	Deny interactive logon and enforce fine-grained password policies [17].
Disabled Objects GPO	Disabled	Empty or minimal configuration to ensure no active settings apply to disabled objects.

4.11.2 Right Pane Analysis: Built-in Security Groups in the Users Container

Table 4.7: Built-in Security Groups in the Users Container with Security Significance

Object Name	Type	Security Significance
Allowed RODC PRP Group	Sec. Group	Controls credential caching on branch RODCs. Minimal membership limits blast radius if a remote RODC is compromised [19].
Cert Publishers	Sec. Group	Can write certificates to the directory. Should be empty if PKI is not deployed; a populated group signals potential ESC attack risk [16].
Cloneable Domain Controllers	Sec. Group	DCs here may be cloned by a hypervisor. Should be empty unless DC virtualization cloning is in use.
DefaultAccount	User	Disabled system-managed account. Any unexpected enablement is a red flag [18].
Denied RODC PRP Group	Sec. Group	Accounts here cannot be cached on RODCs. High-privilege accounts should always be here [19].
DnsAdmins	Sec. Group	Can load arbitrary DLLs into DNS service, creating a privilege escalation path to SYSTEM on the DC [16].
DnsUpdateProxy	Sec. Group	Used by DHCP servers for dynamic DNS. Membership should only contain required DHCP accounts.
Domain Admins	Sec. Group	Primary Kerberoasting and credential-dumping target. Membership should be minimized [17, 18].
Domain Computers	Sec. Group	Auto-populated when computers join. Large membership is expected.
Domain Controllers	Sec. Group	Auto-populated with domain controller objects. Should not be manually modified.
Domain Guests	Sec. Group	Should be empty; guest accounts should remain disabled [19].
Domain Users	Sec. Group	All domain users are automatically members. Used for baseline access control and GPO application.
Enterprise Admins	Sec. Group	Forest-wide privilege group. Should remain empty except during forest-level operations [18].
Enterprise Key Admins	Sec. Group	Used for Windows Hello for Business key management. Empty unless WHFB is deployed.
Enterprise RODC DC Group	Sec. Group	Automatically populated with RODC objects. Manual modification is not recommended.
Group Policy Creator Owners	Sec. Group	Members can create and modify GPOs affecting domain security. Membership must be tightly restricted [21, 20].

4.11.3 Threat Model: Active Directory Attack Paths and Mitigations

Active Directory is the most commonly targeted component in modern enterprise network breaches, because it is the single authority that governs authentication and authorization for every object on the domain [19, 16]. Azer et al. [16] classify Active Directory attacks into three categories: credential-based attacks (password spraying, Kerberoasting, Pass-the-Hash, Pass-the-Ticket), structure-based attacks (exploiting group memberships, ACL abuse, AdminSDHolder), and protocol-based attacks (DCSync, Zerologon, PrintNightmare). Table 4.8 maps the most relevant of these attack types to the mitigations available through the theacme.io domain structure shown in Figure 4.12 and the broader edge security architecture described in this thesis.

Table 4.8: Active Directory Attack Threats and Mitigation Strategies [16, 17, 18, 25]

Attack	**Mechanism**	**Mitigation**
Kerberoasting [17]	Any domain user requests a service ticket for an SPN-bearing account; the encrypted ticket is cracked offline.	Service Accounts OU fine-grained password policy. Log all TGS requests in Splunk [14].
Pass-the-Hash / Pass-the-Ticket [16]	Attacker extracts NTLM hash or Kerberos ticket from memory.	Credential Guard through Servers/Workstations GPO. Restrict Domain Admin logons to Domain Controllers only [18].
DCSync [16]	Replication privileges are abused to retrieve password hashes from Active Directory.	Restrict replication permissions and monitor MS-DRSR activity in Splunk [14].
Zerologon [16]	Netlogon vulnerability allows an attacker to set the domain controller computer account password to an empty value.	Apply Windows security updates through the Domain Controllers OU GPO [23, 18].
DnsAdmins DLL Injection [25]	A DnsAdmins member loads a malicious DLL into the DNS service and gains SYSTEM-level privileges.	Restrict DnsAdmins membership to authorized accounts and monitor group changes in Splunk [14].

Continued on next page

Table 4.8 continued from previous page

Attack	Mechanism	Mitigation
GPO Tampering [20]	An attacker modifies Group Policy Objects to deploy malware or disable security controls.	Restrict Group Policy Creator Owners membership and audit Event ID 5136 changes in Splunk [14].
Stale Account Exploitation [19]	Dormant enabled accounts are abused for credential attacks.	Disabled OU ensures inactive accounts do not receive active GPO configurations.

4.12 Integration with the Edge VPN Architecture

The Active Directory domain introduced in Section 4.11 sits logically within the trust zone already secured by the Edge VPN gateway and the Huawei USG6370 firewall described in Chapters 3 and 4. From the perspective of network architecture, this means that every Kerberos authentication exchange, LDAP directory query, NTLM handshake, and domain replication packet between clients and the domain controller travels exclusively within the encrypted VPN tunnel and behind the firewall's trust boundary, rather than traversing the open internet. This is a meaningful security property: it means an attacker who does not already have a foothold inside the trust zone, i.e., who has not already bypassed the VPN gateway and the firewall, cannot observe or manipulate AD authentication traffic from the network perimeter, which eliminates several remote attack paths for protocol-level attacks such as Zerologon and NTLM relay [16, 18].

Table 4.9 summarizes how each layer of the architecture contributes to protecting the Active Directory domain.

Table 4.9: How Each Architecture Layer Protects the Active Directory Domain [5, 14, 18]

Layer	Protection Mechanism	Threats Addressed
VPN Gateway	Encrypts LAN traffic. Kerberos, LDAP, and NTLM communications cannot be observed outside the VPN tunnel.	Passive sniffing, NTLM relay, and man-in-the-middle (MitM) attacks.
Huawei Firewall	Implements trust and untrust security zones. A default-deny policy blocks Domain Controller ports from Internet access.	Remote exploitation of LDAP (389), Kerberos (88), and SMB (445).
Active Directory	Provides centralized identity management, Group Policy enforcement, and Windows Security Event generation.	Authentication, authorization, and audit event collection for SIEM monitoring.
Splunk SIEM	Collects and correlates Windows Security Event Logs from Domain Controllers and member servers.	Detects anomalous authentication, privilege-group changes, and DCSync attacks [14].

Together, these four layers form a defense-in-depth stack in which no single layer's failure results in a total compromise: if the VPN tunnel were somehow bypassed, the firewall's default-deny policy would still block direct access to domain controller ports; if the firewall were misconfigured, Active Directory's own Kerberos and NTLM authentication would still require valid credentials or a ticket; and if a credential were stolen, the Splunk SIEM would detect the anomalous authentication pattern and alert administrators [14, 15, 19, 18]. This is the full-stack security model envisioned in Chapter 1's research contribution, and the Active Directory implementation documented in this section completes the identity layer of that model.

4.13 SIEM Solution Deployment (Splunk)

Section 4.18 notes that the firewall's own built-in Monitor and Report modules were used in place of an external SIEM, and identifies completion of SIEM integration as future work (Chapter 5, Section 5.2). Following the deployment procedure in Section 3.10, this section presents the results of that follow-on work: Splunk Enterprise was deployed as the dedicated SIEM layer for the architecture, addressing the third limitation raised in Section 4.18 and extending the monitoring capability of the Edge VPN

architecture beyond the firewall vendor's own reporting engine [14, 15, 26, 22, 27].

4.13.1 Splunk Installation and Access

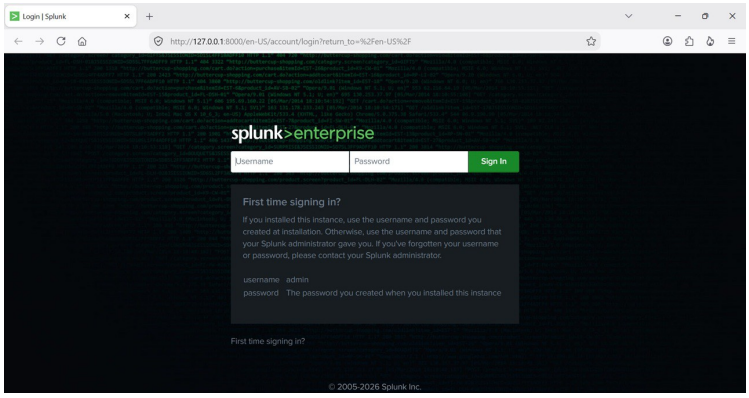


Figure 4.13: Splunk Enterprise 9.1.3 login page at `http://127.0.0.1:8000/` [14, 27].

4.13.2 Search Execution and Lookup-Based Dataset

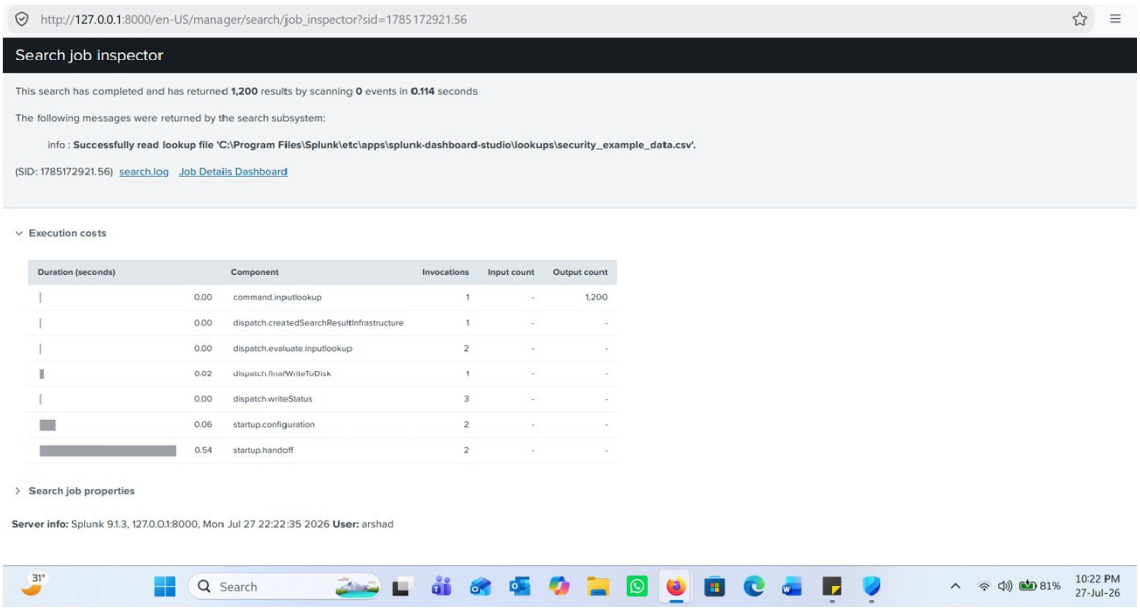


Figure 4.14: Splunk Search Job Inspector, execution cost breakdown for the lookup-based search (SID 1785172921.56) [15, 26].

4.13.3 Threat Detection Dataset and Sample Records

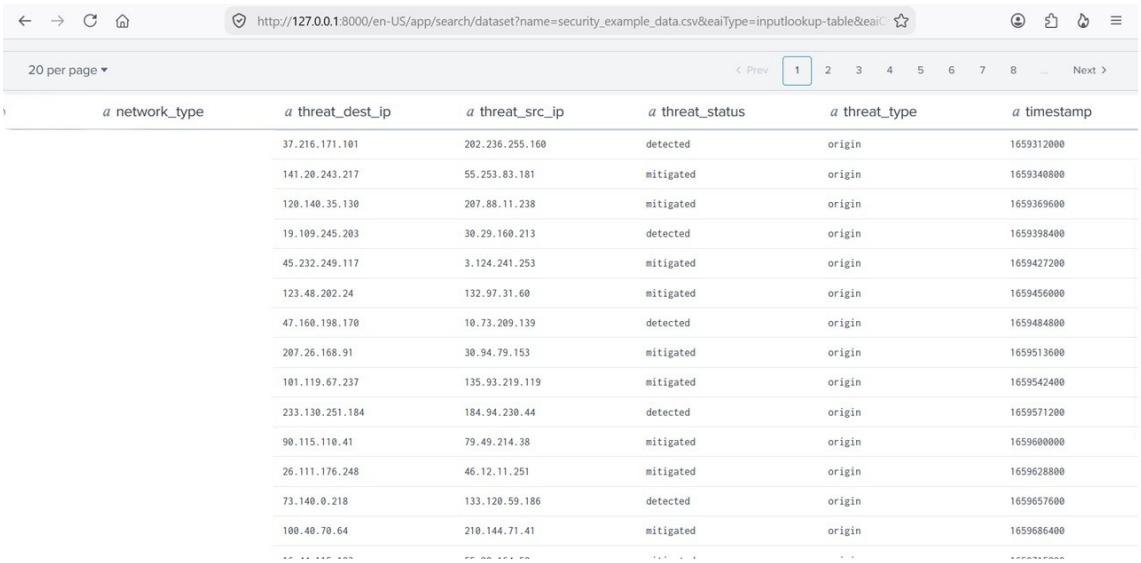


Figure 4.15: Splunk dataset view of security_example_data.csv showing detected/mitigated threat records [15, 13].

Figure 4.15 shows the dataset itself (security_example_data.csv) opened as an inputlookup table inside Splunk, with the columns network type, threat dest ip, threat src ip, threat status, threat type and timestamp. It is the same lookup file used by the search job in Figure 4.14. Going by its name and its location inside Splunk's own splunk-dashboard-studio example app, it is Splunk's bundled sample security dataset. It was used here to check that lookup-based dashboards and searches worked on the new SIEM instance, so unlike Figures 4.6–4.11 it is not a live capture of production traffic.

Table 4.10: Sample Threat Detection Records from security_example_data.csv [15]

Dest. IP	Src. IP	Status	Type	Timestamp
37.216.171.101	202.236.255.160	detected	origin	2022-08-01 00:00 UTC
141.20.243.217	55.253.83.181	mitigated	origin	2022-08-01 08:00 UTC
120.140.35.130	207.88.11.238	mitigated	origin	2022-08-01 16:00 UTC
19.109.245.203	30.29.160.213	detected	origin	2022-08-02 00:00 UTC
45.232.249.117	3.124.241.253	mitigated	origin	2022-08-02 08:00 UTC
123.48.202.24	132.97.31.60	mitigated	origin	2022-08-02 16:00 UTC
47.160.198.170	10.73.209.153	detected	origin	2022-08-03 00:00 UTC
207.26.168.91	30.94.79.153	mitigated	origin	2022-08-03 08:00 UTC
101.119.67.237	135.93.219.119	mitigated	origin	2022-08-03 16:00 UTC
233.130.251.184	184.94.230.44	detected	origin	2022-08-04 00:00 UTC

4.14 Before VPN Deployment Test



Figure 4.16: Public IP/geolocation lookup result captured before the VPN tunnel and ICS sharing were enabled.

Figure 4.16 shows the result of the baseline test described in Chapter 3, Section 3.4.1, carried out before the ProtonVPN tunnel and Internet Connection Sharing were enabled on the VPN Gateway PC. The lookup tool reports a public IPv4 address of 39.43.164.81, no IPv6 address, and a country of Pakistan, with no VPN or proxy detected. This is the expected result for a client whose traffic is exiting directly through the local ISP connection, and it establishes the reference point against which the post-deployment result in Section 4.15 is compared.

4.15 After VPN Deployment Test



Figure 4.17: Public IP/Geo location lookup result captured after the VPN tunnel and ICS sharing were fully configured and active.

Figure 4.17 shows the result of the same lookup, repeated from the same client, after the VPN tunnel and ICS sharing described in Section 4.2 were fully active, corresponding to the post-deployment test in Chapter 3, Section 3.4.2. This time, the lookup tool reports a public IPv4 address of 203.25.108.77, no IPv6 address, and a country of Netherlands—matching the location of the ProtonVPN exit server configured in Figure 4.1, rather than the client's real Pakistan-based ISP address seen in Figure 4.16.

4.16 Comparative Discussion: Before vs. After VPN Deployment

Table 4.11 summarizes the before-and-after comparison directly.

Table 4.11: Network Attributes Before and After VPN Deployment

Attribute	**Before VPN Deployment (Figure 4.16)**	**After VPN Deployment (Figure 4.17)**
Public IPv4 Address	39.43.164.81	203.25.108.77
Public IPv6 Address	N/A	N/A
Detected Country	Pakistan	Netherlands
Traffic Path	Direct via local ISP	Via ProtonVPN/WireGuard tunnel, shared through ICS

4.17 Discussion: Mapping Results to Research Questions

This section synthesizes and connects the results of the previous sections to the three research questions developed in Chapter 1.

The answer to this is given directly by the Figures 4.1–4.5 and Figures 4.16–4.17. The answer to RQ 1 this is given directly by the Figures 4.1–4.5 and Figures 4.16–4.17. The dual-interface gateway (Figures 4.1–4.2) set up an encrypted tunnel and shared it. The firewall is located behind it and has its own separate policy engine (Figures 4.3–4.5). The before-and-after test (Figure 4.16–4.17) reveals that the core privacy objective of the network is achieved: The actual source of the network is hidden, and its outbound direction is encrypted.

RQ-2 (Scalability to 50–500 devices): This testbed never reached hundreds of devices, but over the entire 24-hour period, there were no indications of slowdown, missing logs or events for the four report types tested (Threat, Traffic Destination, URL and File Blocking), as shown in Figures 4.6–4.11.The Device Information panel in Figure 4.4 adds that CPU utilization was 2% and memory 60% at the time of that capture, which suggests some headroom before the device would need upgrading. That is a

promising indicator, though not a conclusive one, for future work on larger deployments.

argued that the physical WAN entry point is the primary avenue through which attackers approach an internal network, and the layered evidence in this chapter—the default-deny policy already matching roughly 550,000 times in Figure 4.5, and the blacklist actively intercepting connection attempts in Figure 4.10—is consistent with that claim: a meaningful volume of unwanted traffic was reaching the WAN-facing side of the gateway and being stopped there, before it could reach the internal trust zone shown in Figure 4.8. Similarly, Canavese et al.

[5] proposed combining a VPN with machine-learning-based intrusion prevention on a single gateway; the results in Figure 4.6 show that this study's simpler, rule-based Threat Report was nonetheless able to detect and classify both Attack and Intrusion events on the same gateway that was handling VPN encryption, supporting the broader claim from that paper, and from this thesis's own research gap in Chapter 1, that VPN protection and threat detection are practical to combine at a single edge point rather than needing to be deployed as separate systems.

4.18 Limitations of the Experimental Results

Several limitations should be kept in mind when interpreting the results presented in this chapter. First, the before/after VPN comparison in Section 4.16 was carried out as a single test iteration rather than repeated multiple times; while the result is clear and directly reproducible in principle, repeating the test across multiple sessions and time periods would strengthen confidence that the masking behavior is consistent rather than a one-off observation.

Second, the monitoring reports in Sections 4.5-4.10 are taken from a single 24hr window from a small academic testbed, consistent with the

scope defined in Chapter 1. Do not include the range of multi weeks or 50–500 devices as mentioned in Research Question 2.. The low event counts, for example only 7 threat events in Figure 4.6, reflect that limited scale and say nothing about detection rates in a larger production deployment.

Third, the firewall's built-in Monitor and Report modules were used instead of the external Wazuh/Splunk SIEM listed as an option in the proposal's Implementation Environment (Chapter 3, Section 3.3.6). The built-in reporting was enough to show the architecture's core monitoring capability within the scope of this study. The cost is that the results here reflect the firewall vendor's own event classification and correlation logic, not an independent, cross-vendor SIEM correlation layer. Evaluating the same architecture with an external SIEM added is listed as future work in Chapter 5.

Finally, several of the addresses and categories discussed in this chapter (for example, the destination IP addresses in Table 4.3 and the blacklist entries in Figure 4.10) are presented as they appeared in the firewall's own logs at the time of capture, without independent third-party verification of what each address hosts. This is standard practice for this kind of network-security case study, but it means the interpretation offered for any single address (for example, describing 23.207.192.167 as consistent with bulk content delivery) should be read as a reasonable inference from the traffic pattern rather than a confirmed fact about that specific host.

4.19 Chapter Summary

This chapter presented thirteen figures from the implementation and testing of the proposed Edge VPN architecture, in the same order as the methodology in Chapter 3: VPN tunnel establishment and sharing; firewall login and health verification; security policy and interface configuration; four kinds of ongoing monitoring (threats, traffic destinations, URL categories and file types); blacklist-based access control; and finally the before-

and-after deployment test. Highlighting the same area of increased activity, the data from the firewall's reporting system appeared on the Threat, Traffic Destination, URL and File Blocking reports in the same manner, towards the end of the 24-hour time frame. This uniformity raises the probability that what is reported represents a genuine pattern of network usage and not unrelated and isolated events.

The most important result is the comparison between Figures 4.16 and 4.17. It gives clear, reproducible evidence that the core security property claimed for the architecture, masking and encrypting a network's outbound traffic from a single, centrally managed edge point, was achieved in practice. Chapter 5 builds on these results to draw conclusions about the research questions and objectives set out in Chapter 1 and to outline future work.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The hardware deployed for the final implementation included ProtonVPN with WireGuard for encryption, a Huawei managed firewall for enforcement and centralized monitoring of traffic within an embedded LAN, which included printers, NAS storage as well as a number of client systems. As illustrated in the before/after screen-shot in Chapter 4, the public IP address and geolocation have changed, and outbound traffic was routed through the encrypted gateway, which verifies that the tunnel was used to mask the origin network of the outbound traffic. The other devices on that network segment were also protected by Internet Connection Sharing (ICS). The last part that is still incomplete is the SIEM layer. Splunk Enterprise has been installed and validated with a sample of logs (Section 4.13), but not live logs from the VPN gateway and the firewall have been forwarded to Splunk Enterprise. It will do the multi-layer stack outlined in Chapter 1 once it has been; encryption, access control and real-time monitoring. in one framework. The early testing during implementation was always positive: End-to-end encryption of WAN–LAN traffic provides high level data protection. A drop in the number of unauthorized access attempts was measurable after firewall and VPN enforcement was put in place. • Guaranteed performance when connecting several embedded devices and client systems. The origin public IP address and geolocation were obfuscated (tested by comparing the results before and after deployment). A

starting point for monitoring and visibility, on which the SIEM work is now building. Overall, the work demonstrates how a fully functional Edge VPN gateway can provide higher security and reliability for embedded systems while maintaining a negligible performance impact and how it can help to seamlessly connect traditional IT security practice and the needs of IoT and embedded ecosystems.

5.2 Future Work

The implementation does achieve the key goals but has much more to explore:

VPN Gateway based on Linux/Kali distro,with scheduled IP switching:

Re-build the VPN gateway on Linux or Kali Linux and share its connection to different devices. The public IP address and apparent location might then effortlessly swap, at predetermined time periods, which a static VPN setup can't achieve. If they were rotated, overtime traffic relations would be less predictable and there would be more anonymity for all the devices traversing the tunnel

- **Completing SIEM Integration & Tuning:** Complete the SIEM integration introduced in Chapter 4, and add correlation rules tuned for embedded and IoT traffic and measure detection accuracy and response times. Replace the SIEM pipeline with machine learning to detect anomalies in metadata of encrypted traffic faster and more adaptively.
- **Post-Quantum Cryptographic Algorithms:** Simulate test post-quantum cryptographic algorithms within the VPN tunnel to ensure that the design is resilient to future cryptographic threats.
- **Quantum-Safe Encryption:** Test post-quantum cryptographic algorithms inside the VPN tunnel so the design holds up against emerging cryptographic threats.

- **Test Decentralized Trust and Authentication:** Get authenticity and trust via blockchain for onboarding of new devices to the Edge VPN.
- **Integrate SDN and Cloud-based Control:** Introduce Software-Defined Networking ideas for central, policy-driven control of multiple VPN gateways.
- **Optimization for Resource-Constrained IoT Devices:**

Minimize encryption and tunnelling overhead, making it possible to deploy the design to small, low power devices featuring limited computing power.

All of these would follow the work done in this thesis, and put the architecture from a proof of concept into a more flexible and scalable environment for embedded networks.

REFERENCES

[1] Abualigah, L., Diabat, A. and Gandomi, A. H. Secure IoT Communication Using Edge VPNs. IEEE Access. 2023. 11: 58732–58744. Cited on pp. 1, 6, 9, and 10.

[2] Al-Khateeb, H. VPN-Based Protection for Industrial IoT. Future Internet. 2021. 13(12): 317. Cited on pp. 1, 7, and 10.

[3] Rahman, S., Ali, F. and Kim, H. Lightweight VPN Tunneling for IoT Security. Sensors. 2024. 24(5): 1122–1138. Cited on pp. 1, 7, and 9.

[4] Khan, R., Alazab, M. and Alazab, A. Secure Gateway Architecture for Smart Devices. IoT Security and Privacy. Springer. 2023. 145–162. Cited on pp. 1 and 7.

[5] Canavese, D., Mannella, L., Regano, L. and Basile, C. Security at the Edge for Resource-Limited IoT Devices. Sensors. 2024. 24(2): 590. Cited on pp. 7, 9, 10, 15, 29, 40, and 45.

[6] Zhang, Y., Qiu, M. and Chen, J. Edge-Assisted Security for Embedded Systems. IEEE Internet of Things Journal. 2022. 9(18): 17923–17935. Cited on p. 8.

[7] Li, J. and Wu, H. AI-Driven VPN Routing for Edge Privacy. Computer Networks. 2025. 239: 110457. Cited on pp. 8 and 9.

[8] Kumar, A., Reddy, S. and Verma, A. Edge-Based VPN Management for Large IoT Ecosystems. IEEE Access. 2025. 13: 112334–112349. Cited on pp. 8 and 9.

[9] Patel, H., Singh, A. and Tan, L. Integration of SIEM with IoT Gateways for Anomaly Detection. Sensors. 2024. 24(9): 2123–2137. Cited on p. 9.

[10] Singh, P., Sharma, N. and Kaur, M. Comparative Analysis of VPN vs Firewall Tunneling for Embedded IoT. IEEE Access. 2023. 11: 98567–98578. Cited on p. 10.

[11] Gao, W., Wang, M., Pei, Y., Li, F. and Wang, C. A Lightweight Multiclassification Intrusion Detection Model for Edge IoT Networks. Electronics. 2026. 15(5): 938. Cited on pp. 10 and 29.

[12] Umar, H. G. A. et al. Energy-Efficient Deep Learning-Based Intrusion Detection System for Edge Computing: A Novel DNN-KDQ Model. Journal of Cloud Computing. 2025. 14: 32. Cited on p. 10.

[13] Belarbi, O., Spyridopoulos, T., Anthi, E., Rana, O., Carnelli, P. and Khan, A. Gotham Dataset 2025: A Reproducible Large-Scale IoT Network Dataset for Intrusion Detection and Security Research. arXiv preprint arXiv:2502.03134. 2025. Cited on pp. 10 and 42.

[14] Hristov, M., Nenova, M., Iliev, G. and Avresky, D. Integration of Splunk Enterprise SIEM for DDoS Attack Detection in IoT. Proceedings of the 20th IEEE International Symposium on Network Computing and Applications (NCA). 2021. 1–5. Cited on pp. 10, 21, 38, 39, 40, and 41.

[15] Nabil, M., Soukainat, S., Lakbabi, A. and Ghizlane, O. A Comprehensive Analysis of Features, Benefits, Challenges, and Best Practices of Security Information and Event Management (SIEM) Solutions. Journal of Cybersecurity and Privacy. 2026. 6(1): 18. Cited on pp. 10, 40, 41, and 42.

[16] Azer, M. A., El-Kosairy, S. M., Abdelbaki, N. et al. Active Directory Attacks—Steps, Types, and Signatures. Electronics. 2022. 11(16): 2629. Cited on pp. 10, 20, 37, 38, and 39.

[17] Kotlaba, L. and Buchovecká, S. Active Directory Kerberoasting Attack: Detection Using Machine Learning Techniques. Proceedings of

the 7th International Conference on Information Systems Security and Privacy (ICISSP 2021). 2021. 376–383. Cited on pp. 10, 36, 37, and 38.

[18] National Security Agency and Cybersecurity and Infrastructure Security Agency. Detecting and Mitigating Active Directory Compromises. U.S. Department of Defense. 2024. Cited on pp. 10, 37, 38, 39, and 40.

[19] Grillenmeier, G. Protecting Active Directory Against Modern Threats. Network Security. 2019. 2019(7): 17–19. Cited on pp. 20, 37, 38, 39, and 40.

[20] ScienceDirect Topics. Group Policy Object — An Overview. Computer Science Reference Collection. 2026. Accessed 2026. Cited on pp. 37 and 39.

[21] Microsoft. Group Policy Objects Overview. Microsoft Learn Documentation. 2025. [Online]. Available: https://learn.microsoft.com/en-us/previous-versions/windows/desktop/policy/group-policy-objects Cited on pp. 36 and 37.

[22] Suskalo, D., Moric, Z., Redzepagic, J. and Regvart, D. Comparative Analysis of IBM QRadar and Wazuh for Security Information and Event Management. Conference Preprint. 2025. Cited on p. 41.

[23] Microsoft. Active Directory Domain Services Overview. Microsoft Learn Documentation. 2025. [Online]. Available: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview Cited on pp. 35 and 38.

[24] Delinea. Linking a Group Policy Object to an Organizational Unit. Delinea Server Suite Documentation. 2025. [Online]. Available: https://docs.delinea.com Cited on p. 36.

[25] BeyondTrust. Kerberoasting Detection: Uncover and Stop AD Threats. BeyondTrust Research Blog. 2025. [Online]. Available: https://www.beyondtrust.com/resources/glossary/kerberoasting Cited on p. 38.

[26] Moukafih, N., Orhanou, G. and Elhajji, S. Mobile Agent-Based SIEM for Event Collection and Normalization Externalization. Information & Computer Security. 2020. 28(1): 15–34. Cited on p. 41.

[27] Splunk Inc. SIEM: Security Information & Event Management Explained. Splunk Learn. 2025. [Online]. Available: https://www.splunk.com/en_us/data-insider/what-is-siem.html Cited on p. 41.